Still Alive With Sir Clive!

ZXir QLive Alive!

The Timex/Sinclair NorthAmerican User Groups Newsletter

Volume 5 Number 2 Summer '95

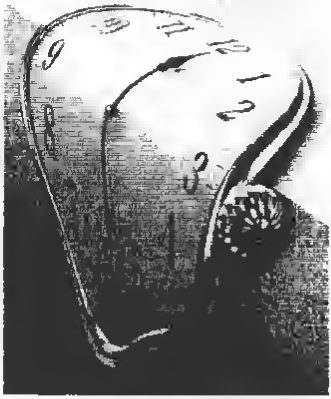
Chairman Donald S. Lambert Auburn, IN

MEMORY MAP

ROUTINES

FILES

SUBROUTINES



Timex DATA LINK Wrist Watch
Download info from your computer to the watch; appointments, phone list, to-do list, etc. Hold the watch in front of your blinking monitor while running the Windows software, the screen flashes transmit the data to the watch — TS-1000/2068 tape LOAD?

ZXir QLive Alive! ©
Established 1991 The Timex/Sinclair NorthAmerican User Groups Newsletter

T/SNUG Information

T/SNUG

Here is the list of T/SNUG Chairmen and how to contact them. We wish to support the following SIGs: ZX-80/81 TS-1000, SPECTRUM, TS-2068, TC 2068, Z88 and QL. If you have any questions about any of these fine machines contact the:

Chairman

Chief Motivator
Donald S. Lambert (ISTUG)

Vice-Chairmen

Tape & JLO PD Library
D G Smith
R 415 Stone St.
Johnstown, PA 15906
814 535-6998

Z88 Library
Dave Bennett (HATSUG)
329 Walton St. Rear
Lemoyne, PA 17045
717 774-7531

ZX-81 PD Tape Library
Ed Snow
2136 Churchill Downs Cir.
Orlando, FL 32825
407 380-5124

RMG Enterprises
Rod Gowen (CCATS)
14784 S. Quail Grove Cir.
Oregon City, OR 97045
503 655-7484 FAX 503 655-4116

TS-2068
Rod Humphreys (VSUG)
10984 Collins Pl.
Delta, BC V4C 7E6 Canada
604 583-2819

QL PD Library
John Donaldson (CATUG)
835 Foxwood Cir
Geneva, IL 60134-1631
708 232-6147

BBS — GATOR
Bob Swoger (CATUG)
613 Parkside Cir.
Streamwood, IL 60107-1647
708 837-7957 Work 708 576-8068

Editor/Treasurer

LarKen PD Library
Abed Kahale (CATUG)
335 W Newport Rd.
Hoffman Estates IL 60195-3106

ZXir QLive Alive!

Is the newsletter of T/SNUG, the Timex/Sinclair North American User Groups, providing news and software support to the T/S community in a **volume** of four newsletters per year; beginning with the Spring (March) issue.

T/SNUG's main goal is to keep our Magazine, our vendors and our repair service alive for the benefit of T/S users.

These valuable services shall have free advertising space in this user supported Newsletter so that they can see that we are still active out here. We must support their services whenever possible

Another T/SNUG goal is to unearth titles of all known Public Domain and commercial software available for all Timex/Sinclair machines, building a library and providing lists of that software showing both the source and the availability

If you have solved a problem or you have a problem in one of your software or hardware, please share it with the rest of us.

Trea$ury Note$

As of May 26, 1995, we have a balance of $1182.82

You can keep T/SNUG alive by an annual contribution of $12 for one **volume** made payable to Abed Kahale Send check to -

ABED KAHALE
335 W NEWPORT RD
HOFFMAN ESTATES IL 60195-3106
Phone - 708 885-4337

Back Newsletter copies are available for $0.50 each postpaid

Article Contributions

Send in your articles by tape or disk and your inputs to:-

DON LAMBERT
ZXir QLive ALive! Newsletter
1301 KIBLINGER PL
AUBURN IN 46706-3010
Phone 219 925-1372

Or by hardcopy to:— Abed Kahale

GATOR's TWISTED PAIR

We have a 24 hour BBS and encourage you to exchange mail and contribute to the Upload Section. Use it and have fun!' (8N1 300-2400 BAUD)

Call 708 632-5558

and Register using your first name, last name and phone number along with a password you won't forget, and ***Write It Down!*** Do not try to do anything else this first time because all the board options will be locked-out.

When you call-in the next time, you will have Level 5 security and be able to enjoy full user privileges. The BBS has smaller sections called conferences. Select "J" for "Join a Conference" to see the different user groups Select "TIMEX" to get into the Sinclair Section The mail you then read will only be from other TIMEX Sinclair users but all SIGs share the same bulletins. Use extension .ART for articles, .ADS for ads and .NWS for news when uploading.

For help, contact the SYSOP by leaving a message, mail, e-mail or phone Bob Swoger

SYSOP --==GATOR==--

Welcome

Gertie Anderson
James Curry
Ed De Boer
Ted Heckman
G. David Johnson
J. O'Donnel
Paul Robinson
Larry South
Jose Vasquez
Basil Wentworth

Input/Output

by Abed Kahale

I think it was a good decision to move your publication date forward so that ZXir QLive Alive! and UPDATE! aren't published in the same month. Since our publications are dwindling it is nice to have them coming spaced out. You might consider trying to mail half way in between UPDATE!'s

Les Cottrell
Cocoa, FL

Thank you for letting us know that we made the right decision. We will consider moving it forward again but, it all depends on how much material we have on hand at the time as you can appreciate.

Enclosed is a disk a sort-of-reply article to Bill Harmer's TS Bulletin. I don't think it will start too many sparks.

I have enclosed a copy of DME the Atari ST editor that was ported to QL too

Hope to see you in Oak Ridge for the QL show. I know I will see you here in Dayton for the ComputerFest (and the free burgers -))

Happy hacking,

Tim Swenson
Heuber Heights, OH

I am sure Bill Harmer will appreciate your views on the subject and thank you for the invite.

Thank you for such a great source of information on the Timex machines, especially the TS-2068, for which I own and use weekly. Thank you and keep up the good work!

Robert Gilbert
Waltham, MA

Words like yours keep us going.

I really appreciate your hanging in with the Sinclair computers. Although I use IBM compatibles at work, the Z88 is still my favorite workhorse!

I've held onto my TS-2068's but have been too busy to work with them, as soon as I find some free time, I plan to return to the fold.

I apologize for the delay in paying for the subscription. Is it possible for you to modify your labels to include the expiration date? (for those of us whose ***to-do*** list would require a 36 hour day)

Francine Sklar
Loch Sheldrake, NY

No need for apologies, you really were on time - February. You're doing just fine with the 24 hours day young lady. I use to have the expiration date on the labels but we decided (as you might have read in the Summer '94 issue, page 4, Policy Declaration) to go to membership by volume of four issues beginning with March (Spring) of every year as was originally intended. Appreciate your concern and thanks for the good word.

Here is my annual contribution to T/SNUG, keep up the good work. I always look forward to reading ZXir QLive Alive! newsletter. 73es

Harry Miller W1DRD
Berlin, MA

We'll do our best, and keep on DXing.

Thank you, for supporting the continued usage of the delightful little Timex-Sinclair 2068. Especially, we appreciate the advance warning of the Newsletter's change in schedule. Keep on TIMEX'n.

David Lassov
Tucson, AZ

And thank you for your contributions.

Enclosed my check for '95 T/SNUG membership. Fortunately I am combining some spring cleaning while I pack for Spain I came across a reminder - *Spain only for 2 wks.*

I am like a newborn - new Z88 arrived 1/4/95, I crashed it 2/13/95 and now longing for a Z88-style BBC BASIC manual. Sshh - doing what seems right, so, I am sort of teaching myself BBC BASIC. There seems to be bug in my screen editor program - CLI (Command Line Interpreter), trying to execute that program in the Z88 manual may be what crashed my Z88. I did not know who I'd been or who I'd planned to be for 3 days afterwards, till I got my diary plumped out again with warning of impending debts, etc. May be CRASH was a blessing. I also lost lots of chores I'd planned for my self, I rescheduled and rescheduled. I thought I had learned my lesson with *"The Worx!"*. Ain't technology great? I used to keep

my plan for the day on a scrap of paper and just wad it up when I got hopelessly behind.

I am also fretting a way to transfer material from Z88 to 2068 I guess I have to jure up modem or 2, (if I had a serial connector with 9-pin plug on each end) but clang, that seems cumbersome Seems like I could pipe it thru 9-pin on LarKen, or serial port on Zebra controller, or may be joystick 9-pin gidget on left side Androids prog for 2068 accepted noise commands, I thought (back 10 years since I played with that) - Anyhow I thought left joystick port input different in some way to right. Regards,

Joan Kealy
Brackettville, TX

John J Shepard, is one that I know off that has the experience in transferring files between the Z88 and the 2068 — by modem Not having a Z88, I would venture to say that if both computers have serial ports, then all you need is a cable with one end having the connector pins in reverse order from the other connector The joystick port won't do. You can only transfer text - ASCII files Bon voyage

A member writes,

I know very little about electricity and I need to know what gauge wire I should use to carry up to 4 amperes (amps) at 5 and 12 volts for a distance of 4 5 feet? It will be much easier to run a flat cable than four separate wires to power my disk drives.

You can use the cable with no problems carrying 4 amps maximum The drives are only 'on' intermittently For signal wires, such as to your printer you can use up 6 feet of cable

The best way to find out whether you have the right wire size for any electrical use is - If the wire or cable gets warm to your touch while you have the maximum load or current through it, then the wire gauge is too small and you should go to a larger size

To give you a guide Hold on to the rubber cord of your toaster while you are making toast, this is the maximum safe wire temperature allowed by Underwriters Laboratory Inc You should not exceed this cord temperature

UL Inc is an organization created by manufacturers for the safety of the consuming public It is NOT *the wolf minding the sheep* Manufacturers through out the industry (foreign or domestic) pay UL to test their electrical/electronic products, intended for home use, for compliance with the safety standards created over the years by UL to protect the consumer the public against all kinds of hazards

While it is not a government agency, foreign goods have to comply with these safety standards or else their products may not be imported by a US firm

If you look at an appliance, monitor audio equipment, washer dryer computer, toaster, refrigerator shaver, light fixture, etc you will normally find an identification label carrying either the UL insignia or something to the effect of Listed Recognized or Approved by Underwriter Laboratories Inc

Unfortunately, the auto manufacturers never joined in with the rest of the industry

I don't want to bore you with details, but if there is interest let me know I am familiar with most of UL standards

You only need to learn one formula very well to become proficient in electricity, Ohm s Law formula **E = I R**

E (volts) = I (amps) X R (ohms)

of course I = E/R and R = E/I

Also W(watts) = E X I or W = P X R

My toaster is labeled 900 watts 120 volts So I = W/E, 900 divided by 120 = 7 5 amps is the current (I) going through the cord at 120 volts To save money, manufacturers keep the cord length to less than 6 feet so that they don't have to use a heavier gauge cable.

Your 4 5 feet cable has a resistance of 0 29 ohms carrying 4 amps which is W=I²R or 4² X 0 29 = 4 6 watts of heat generated in the cable. Since both drives will not be running at the same time, you won't *feel-the-heat* so-to-speak from the cable wires As you can see *volts* are not part of the formula So voltage is immaterial when figuring wire gauges

The member response;

UL is strictly for safety not for workability for close tolerance applications. Safety takes the first and most important factor Maybe my question was not very well thought out. What I am interested in is more like a chart that will list the amps that various gauges of wire will carry without going out of operating voltages under loads It isn't a question of the wire getting warm, it is a question of whether the wire will deliver the voltage within IC tolerances

Of course maybe what I am worrying about is immaterial but when you pick up a 100 ft. extension cord and they say that you can run up so many amps if the cord is 50 feet it is much higher current. But again I think it is a case of UL and the heating of the cords According to the formula E=I X R and I = 4 and R = (4 5 X 0 29), 4 X (4 5 X 0 29) = 5 22 volts Of course my electronics is very rusty but I under-

stand that to be the voltage drop in the wire So what does actually get to the end of the wire?

You have 4 wires having a resistance of .29 Ω that is to carry a maximum 4 amps Since only 2 wires will carry 4 amps. for one drive at a time, the resistance of the wire is 4 5 X 0 026 = 0 117 The voltage drop is I X R = E which is 4 X 0 117 = 0 468 volts (in round numbers) 5 - 0 468 = 4 53 *volts (the voltage should not go below 4 3 volts)* and 12 - 0 468 = 11 53 volts is the voltage delivered to your drive at maximum current. (If you measure the current, it will be less than 4 amps) UL Inc standards are based on just these calculations and from them they derive a *'temperature rise'* above room ambient temperature of 68°F (20°C) It is not safety hogwash.

I never worked for UL by the way. I did have to meet their standards though To meet them, I had to learn all about them

Since I originally didn't want to go deep into the subject, I did present you with an alternative - the toaster cord Extension cords current carrying capacities are also based on the wire gauge and length too If a 25ft cord can carry 16 amps , a 50ft cord of the same wire gauge can only carry half as much; 8 amps. Wires gauges are specified in ohms per foot among other specs **There is no wire gauge table for current** (amps) There is a guide for electricians on house wiring in the National Electrical Code that cover gauges from #16 to #0000 for 8 to 225 amps This is of no help to your electronic question and will only serve to confuse the issue Here is a list of resistance per foot from the American Wire Gauge (Brown & Sharp) for Standard Annealed Copper Wire -

Gauge	Ohms/ft
#16	0 006
#20	0 01
#22	0.016
#24	0.026
#26	0.041
#28	0 065
#30	0 103
#32	0 164

By the way, there is no such thing as precise as in mechanical precision tolerances in electronics, nothing like ±0 000015√ ±√ microns nor microinch Electronic components have tolerances in percentage Generally resistors have 1% to 10%, capacitors 5% to 20% etc for instance

WOOD & WIND

Thanks for putting information about my business in the last newsletter Keep up the good work. If possible please include the following Ad on a continued basis You can photocopy or redo it.

Whatever is easier for you.

Bill Cable
Cornish, NH

I decided to redo it just for the fun of it Welcome aboard Bill

I got the Spring '95 issue of ZQA! a little while back, and noticed your regular request for article submissions. At about the same time, I was experimenting with my new PC8300 pertaining to programming some music, and I wanted to write myself a "note to file" as a reminder of what I learned. I decided to put the two together, and the result is attached. Rather than sending a manuscript and requiring you to type it in, I thought I could make things easier if I put the text in final form So, I tried to match your page layout, typeface, etc , as much as possible, and (if you want to use the article) about all you really need to do is put the page numbers at the bottom, I realize the PC8300 is a pretty obscure topic in the ZX/TS world, and that programming music on the PC8300 is more obscure still But at least, it's original!

Gilham Parrish
Beggs, OK

Original it is We appreciate your contribution especially in a final form You really didn't have to bother with the formating Thank you

Would like you to send me a complete set of information for the LarKen Disk Interface by Les Cottrell in the Summer 94 issue

I just purchased Don Lambert's TS-1000 LarKen and all is operating well

Naturally I would also like a LarKen for my TS-2068 I would appreciate any additional helpful information Presently I don't understand the need for both the IF and the dock boards

Ferdinand Gunther

The LarKen System is made of a Disk Drive Interface board that fits on the back and a Disk Operating System (DOS) board that fits in the dock (cartridge slot) It is an excellent

system I wouldn't be using my TS-2068 without it See Larry Kenny's description enclosed

Now for the bad news Les Cottrell made the DOS board and got it to work but to my knowledge he has not made the Disk Drive Interface as yet, nor anyone else as far as I know it is a larger board and is more complex with feed-thru connectors which are hard to come by Of course you will need both boards for the system to operate By the way, The dock board will work with the Oliger system

My advice would be for you to buy a system while they are still available See RMG and Mechanical Affinity ads

---

Thank you very much for sending the information you have on the LarKen Disk Interface I now understand that the DOS plugs into the cartridge slot and that the interface board fits on the back of the computer

Under the circumstances I decided to purchase a used AERCO disk drive system from Mechanical Affinity

I needed your help to come to this conclusion I am very grateful for this important information.

Timex/Sinclair Alive and Well.

Ferdinand Gunther
Moses Lake, WA

I am totally unfamiliar with you. I heard that you distribute a publication about TS-1000 PC. I would like to see this item.

Send me as many back issues of your Mag, as $5 can buy and information about subscriptions

Thanks, hope to hear from you soon.

GREETINGS FELLOW ZX TRAVELER.

CARL JONES
16104 W. COURSE DR.
TAMPA FL. 33624

Let's fill in some of the blanks, left in the Appendices of the USER MANUAL for the TS-2068 Personal Color Computer

In Appendix A (Review of TS-2000 BASIC,) there is no mention of the SCROLL function (or operation or whatever) Just do a RAND USR 2361 What happened ??

As for Appendix B — The Character Set, we have Character "INV VIDEO" for code 5, Character "CAPS-SHIFT/SYMBL-SHIFT" for Code 14, and Character "GRAPHICS" for Code 15 Lastly, we use Character " " (null) for Code 0 This is very handy for checking a string X$, to see if it's empty (CODE X$=0)

Consider Appendix D — The System Variables One way of exploiting the built-in routine for scanning the keyboard is, by reading the ASCII CODE of the last pressed key (LAST_K at 23560) after the value of FLAGS at 23611 goes from 220, to some number greater than 220 Another way to scan the keyboard uses INKEY$, right after PAUSE 0 — for example, PAUSE 0 LET C$=INKEY$ Also, the cursor letter goes from C for CAPITALIZED input to K for lower-cased input, as bit 3 of FLAGS2 at 23658 goes from 1 to 0. Finally for Appendix D, the value of PEEK 23688 can be negated with impunity, still giving the *33-column number for PRINT position*. Same goes for the value of PEEK 23689 (either PLUS or MINUS).

Before leaving Appendix D, we have to admit to passing on some erroneous material. POKEing 23692 (SCR_CT) with 0 does not disable the SCROLL operation (which would probably be impossible to disable, anyway) ***but rather*** sets the SCRoll Count to 256.

Please, see Page 29 of CCATS' "the best of THE PLOTTER", There is found Jack Armstrong's article "Tips On Use Of Color Commands In Extended Mode" He shows how to imbed all the colors in the line itself, but never got around to the FLASH command. However, see page A3 of Bill Pedersen's disassembly of 2068 ROM for that and more !!

Here is some interesting CODE for reading the current disc drive number from LKDOS. Big deal ?? It is, should you choose to switch disk drives, but need to remember the original disk drive (automatically, of course)

We will go through the code, which is designed for LarKen DOS Line 10 prepares location 8200 in LKDOS to receive the address of Dvsel, kind of the current drive selected. Line 20 reads this as Z=USR 110 But, we only have Z=2 for drive 0, Z=4 for drive 1, Z=8 for drive 2, Z=16 for drive 3 and Z=128 for drive 4 (RAMDISK) Thus, we need line 30, which transforms Z and allows for numerical rounded off errors Line 40 then corrects for the non-linearity in the equation for drive 4, finally yielding PD as the "Program Disk"

```
10 RANDOMIZE USR 100 POKE 8200 0 8195
20 LET Z=USR 110
30 LET Z=INT (0 5 + LN Z / LN 2) - 1
40 LET PD= (Z AND Z < PI) +(4 AND PI < Z)
```

David Lassov
Tucson, AZ

FROM THE CHAIRMAN'S DISK

Donald Lambert

It took time, a lot of time, but finally SNUG has been put to bed with the money distributed. I know how much there was but I don't know how much went where and I will let those that handled the money state that part. We cajoled, begged and pleaded and finally the money moved from Florida and again we worked to get the distribution finished. How much of my own effort contributed to the distribution, I do not know and I won't even guess at. *But it got done regardless of who did what!* I do thank everyone that had a hand in the process

There is action in the T/S community Sometimes the action becomes known to me. There are still those out there that are wanting more T/S equipment, software and documentation. I write letters to try to point them in the right direction. Occasionally, I do help someone; that is I hear from someone that tells me I helped them. That makes me feel that I am not working alone or writing to a black hole

CARL JONES
15104 WEST COURSE DRIVE
TAMPA, FL. 33624
813-960-2674

One person asked about the A & J stringy floppy for the ZX-81. I had no knowledge of anyone with one, let alone someone that wanted to sell the system But I did have a CAI (Computer Assisted Instruction) stringy floppy so I got it out to test to see if it still worked. I do believe that I spent more time with it than when I first got it It works, it is faster than cassette but not anything like a disk system.

I also had a request for some software for the ZX-81, so I checked that out to be sure it would LOAD and it did. But I had to get into my Load Aid and the Transformer Load Aid to do the LOADing successfully

Another person bought some books and asked about the Load Aid that I had made I wanted to send him copies but the originals were run by a printer that did not have anything but draft quality and that would not reproduce Besides, I had to rewrite and re-edit the material to reduce the wordiness and got it correct (I think), sent him copies of that with the book. Incidentally, postage is getting more expensive for packages

I also ran into an article that I wrote back in 1989 as I worked on a battery backup for the ZX-81 A 12v 2 5 AH gel-cell powered a ZX-81 for over three hours The battery just powered the ZX-81 not anything else. I am working to get that article retyped for later use if it is deemed worth publication.

I have found that projects are sort of like opening Pandora's box. You never know what problems you face till you start. Sometimes the problem is not really a problem - that is - it is a matter of not understanding how to do something and you thought it should have been done a different way Some problems are not solvable so you have to know when to quit. Or to give it to someone else to solve

On the Timex Operating System "Zebra/Portuguese floppy disk system" the drives are FORMATted as 40-track single sided only or 80-track double sided only I had mentioned to Jack Dohany that I thought I would use a 5 25" 40-track single sided drive to be able to use the cheaper disks and a paper punch to make *flippies* out of the 5 25" disks Also an 80-track double sided drive to have that capabilities on 5 25" disks When Jack sent me a 64K controller for the TOS he included a pair of half height 5 25" drives, one was single sided 40-track and the other was a double sided 80-track drive I always test newly acquired drives with the Oliger disk system since I can use the disk drive test program to check spindle rpm The drives LOADed with a little difficulty They FORMATed and SAVEd and could read their own disks But the 40-track drive had a spindle speed of about 286 or 287 and the 80-track had a speed of about 290 When I checked the disks on drives with the correct spindle speed the LOADs were not accepted.

At the time I was only interested in the 40-track drive, so I looked it over and there was no pot that would adjust the rpm. I called a disk drive repair service in Texas (800 # of course) and learned that 1) If there was no speed adjust pot that the drive could not be adjusted 2) Atari was noted for using drives with an rpm of 288 I junked the drive. There is a pot, but it is located where a major disassembly of the drive is required before you can get at it. If I had not destroyed part of the drive before I discovered that, I might have tried to adjust the rpm. I have, however, a double sided drive to use there. The 80-track drive had a pot and adjustment was a breeze.

I had sent Joan Kealy a package of 3" disks (some with programs that she might not have and the rest FORMATted) so that she could send me some of her programs. The disks came back, I did try one and the programs LOADed fine. When this newsletter is put to bed, the part I am working on that is, I will get back to it. One of the things that I need to do is make a case for the disk drives that I plan to use in it. I have plans to have one 3", one 3.5", one 5.25" 40-track and one 5.25" 80-track drive in the case. That means that I will have to figure out the locations of the mounting holes of the drives and the actual drive sizes so that I can mount the drives neatly. There is literature that has that all listed, so when I get that all found I will supply the dimensions for anyone else who wants to make a drive case. In my plans, the drive case is just that, the power supply will be located elsewhere. That is a to-do project when the weather stabilizes so that I can work in the garage. Today is a little on the cool side and is raining. Not a day to cut the case to size outside or to cut the grass.

CAI/ESF Stringy Floppy

by Donald Lambert

It comes with 1 or 2 drives an is menu driven with easy keyboard commands. You can SAVE and LOAD programs at 1100 BAUD on microcassettes. The CAI/ESF is used with the CAI/O board.

The predecessor to MDV

From "The Timex/Sinclair Directory" by E. Arthur Brown Company, 1983

The "Stringy-Floppy" mass storage device is probably the only system available in the US that allows true disk-like capabilities. The CAI/ESF allows you to access specific data without LOADing everything into RAM. It operates on a continuous loop of magnetic tape at very high speed so that information retrieval is not only possible, but fast.

In order to access specific data files without LOADing everything into RAM, a computer must have the capability to selectively read outside data. The TIMEX-Sinclair doesn't come equipped to do this. However it can do it through the proper interfaces. And that is what CAI has done with their 'Stringy-Floppy' system. The interface is called the CAI/O board. Since you'd have to have some sort of disk interface to connect up to a disk drive anyway, the CAI/O board in no way diminishes the value of the stringy floppy. At this writing, there are no disk drives that I know of that come near its price of only $119.95. The CAI/O board sells for $79.95. *(Incidentally, the CAI/O board connects up to a lot of other things as well, like printers and modems.)*

The Stringy Floppy is part of a new generation of mass storage devices manufactured specifically for the T/S computer. It lets you access your programs and data at near the speed of a floppy disk. For example a 10ft. tape cartridge holds 15K of data and can be LOADed in 10-15 seconds. For comparison, a standard cassette player operates at about 250 baud (bits/second) while the Stringy Floppy operates at 11 000 baud. Unlike the sensitive volume and tone controls of a cassette, the CAI/ESF is preset to communicate only with the T/S computer. It LOADs perfectly every time.

Since the Stringy Floppy connects to the CAI/O board on the back of the computer, it leaves the cassette player ports available for ease of program transferring from cassette to Stringy Floppy. You won't have to worry about incompatibility with all of the software currently available on cassette tapes.

I mentioned before that it has the ability to selectively search and retrieve data, it can do this while a program is running as well. This gives the T/S the capability of computerized letter generation. More specifically it can run a word processing program for a form letter, access names, addresses and other information from a mailing list, and fill in the blank, to produce a personalized looking letter. Of course all of this depends on the availability of compatible software.

The Stringy Floppy is menu driven. A menu is displayed on the TV screen giving you LOAD, SAVE and FORMAT options. It's all operated directly through the computer keyboard. You can set up the CAI/ESF to bring programs into the computer while RUNning - no need to enter 'RUN'. Tape loop cassettes are available in 5, 10, 20, 25, 35 and 50 foot length taking approximately 1 minute to cycle completely. The system is also capable of handling two drives increasing memory and speed of data access. The CAI/ESF comes with 2 tutorial programs and a manual.

That was in 1983 when disk drive prices were very high. Even the disks themselves were on the order of up to $2.00 each for 5.25" disks while the drives were close to $200 each plus the disk interface. That was one reason that the T/S computers were so slow to have disk drive interfaces available, those that went to disk drives had money to spend.

The CAI/ESF uses a special tape cartridge called a wafer. The wafer is 2 21/32" long by 1 9/16" by 7/32" thick. The tape itself is 1/16" wide and like the tape in an 8-track audio cassette is endless.

To call the CAI/ESF into use you key in the following.

PRINT USR 10246 (ENTER)

And you get the menu:

ZX-81 ESF MENU
1 LOAD
2 SAVE
3 CERTIFY
4 BASIC
5 SELECT DRIVE

The ESF SAVEs to a file number. It does not use file names so you have to keep a list to be able to know what is on a wafer. It does not assign file numbers as you SAVE but you supply the number and if you reuse a number it will overwrite the previous data under that file number. If you had five files on a wafer and reSAVEd the 2nd file (#2), I don't know exactly what would happen if the reSAVEd file was longer than the first but I suspect that part or all of the third file would be overwritten. There is no wafer directory nor minimum file size on the wafer.

The operation is rather simple and surprisingly reliable. I will take you through the steps to start out with a brand new wafer. Like a disk for a floppy drive the wafer has to be certified (FORMATted). From the CAI/ESF menu you select 3, the computer will ask for a file number, just press ENTER. The screen will go blank and the motor light will come on. Then when the tape reaches the *foil joint* the other light will come on. The FORMATting is verified and the total byte count is displayed on the screen above the menu. If a specific file number is given, the FORMAT (certify) will start at that file number and certify to the end of the tape.

The only problem was that I found two wafers that self destructed, that is the tape parted at the splice/foil and tangled up inside the CAI/ESF mechanism. I had to disassemble the unit on one tape to get the tape out. In the other the tape did not get tangled-up inside the mechanism. In both cases the wafers were no longer usable. The wafers that self destructed were solid black and had labels from A & J. Since I have tested all the wafers (except for some new ones still sealed in plastic bags) there should be no more problems. It was amazing how few moving parts were inside the CAI/ESF.

When any other wafer option is selected the computer will ask for file number. And when ENTER is pressed the CAI/ESF goes into action. While I did not test it, I believe that the CAI/ESF does not really know what order the files are in. I think that at the end-of-file (EOF) marker there is a way for the CAI/ESF to determine that here is a certified space to put a file. If the SAVE is too long, it tries to SAVE on the foil splice, there will be an error message to that effect. In the SAVEing process, it seems to go through the tape to get the foil marker.

When you LOAD a file number and that number does not exist on the wafer, the CAI/ESF goes through the tape from foil to foil before reporting that the file can't be found. However if you have LOADed a file and then LOAD the next file there is almost no elapsed time between the pressing of ENTER and the appearance on the screen of a successful LOAD.

The wafers can be write protected by removing a white dot. Somewhat like the 3.5" disks, to uncover an opening that write protects the recording medium. Just the reverse of the 5.25" disks.

The CAI/O is powered from the computer but the ESF is powered by a 12 VAC .93 amp. transformer. It has the ESF plugged into the rear of it while the right side of the CAI/O has the standard Timex port connector.

There are instructions for program chaining and a chapter in the 16 page manual for machine code buffs. For instance "WEOF 280F ENTRY A=File number. Used to write the EOF marker."

The only real advantage over cassette is that it is so much faster. With the available LOAD AIDS the dependability of a person's own system is great. The undependability appears only when you try to LOAD cassettes from a different machine.

There are programs to accelerate the baud rate of the SAVEing and LOADing routines. The CAI/ESF is an interesting concept that just didn't make it like 8-track cassette systems.

A Word to the Wise

by Tim Swenson

Bill Harmer's article in the Spring '95 issue of ZQA causes some old thoughts to resurface on the future of Sinclair computing, the demise of various User Groups, Newsletters and the move by Sinclair users to other platforms.

The moaning over the demise of Sinclair computing has been droning on for a few years. This is especially apparent in the realm of the TS-2068 and ZX-81, since the amount of new software and hardware is almost nil.

I have come up with a couple of 'rules' on the subject.

1. All computing platforms are essentially the same.

Give me any computer, be it MS-DOS, QDOS, UNIX, etc., and I will be able to do the basics of Word Processing, Spreadsheets, Data Base programming, etc. Granted each platform has some differences over the others, but the basics of being a computer is always there. As the owner of 53 computers, I should know this. I have a QL, an MS-DOS PC, an Atari ST, and a Z88 all set up for use. I have an Epson Geneva that I would like to use more. I have a CP/M machine that my wife used when I took the QL out of town.

2. Your computing platform only needs to change when your requirement changes.

If a CP/M (or 2068) machine met your requirements in 1984, why does it not meet them now? Have your requirements really changed? Barring hardware failures, I can take a CP/M machine and do all of the computing I need to do (mostly word processing and programming).

If you only need your computer to do a few simple tasks, why upgrade to a full-blown Pentium? Look at what your true requirements are.

Plus, output really depends only on the printer and not the platform. If you hooked up a ZX-81 to a laser printer, no one would be able to know that that printout came from a ZX-81.

3. Don't get involved in computer/OS bashing.

We each have our preferences for computers, just as we each have preferences in music, food, and cars. There is no need to justify your computing platform by denigrating another computer platform.

I personally love the QL. I use MS-DOS Windows at work. I have spent a couple of years delving into the depths of UNIX. I have used a variety of operating systems and personal computers. I chose the one I liked best for the price I was willing to pay. I do not need to justify this choice to anyone.

4. Don't alienate those that are leaving Sinclair Computers and moving on to other platforms.

Staying with an orphaned computer has its costs. The hardware will fail (esp. with parts getting harder to find). Getting any new software is getting harder. Support is almost non-existent. It's limited to getting help from other users. The lure of the new platforms with all that new software and support is quite enticing. I mean support in that you can walk into any computer store and buy MS-DOS/Windows programs. It's been 10 years since you could do that for a Sinclair computer.

As people leave the hobby, let them know about the emulators for MS-DOS, Amiga, Atari, and Mac. Let them know about the QXL card. Let them know that they can still have all that new stuff, plus keep some of the old stuff too.

I would like to stay with the QL for many years to come (10-30?). Eventually my hardware will fail. Using new hardware (like the QXL) will allow me to continue using my QL/QDOS. The ZX-81 and Spectrum emulators allow you to continue using what you like without having to worry about your hardware dying. Plus, with the speed of the new PC's, you can have a really souped-up up ZX-81. I would love to see someone buy a 486 PC just to use it as a ZX-81 clone. No Windows at all. Just have the AUTOEXEC.BAT file boot up directly into the ZX-81 emulator.

What it all boils down to is this. Some ZX-81/2068 users have done all that they can on these computers and want to move on. Some still feel that the ZX-81/2068 still suits their needs and want to keep it. The number of new innovations for each computer is fast approaching zero. If you love either computer, stay with it and be happy. Let's hope that the people that have just discovered either computer will breathe new life into the Sinclair community.

of waiting to load a spreadsheet or an AutoCad that takes a long time to learn. After you just begin to learn how to use it, you start discovering its quirks and then you have to learn how to go around them and all you want to do is balance your check book.

Being an amateur programmer who likes to keep up with the state-of-the-art when I can afford it, I found out that every system had its idiosyncrasies that you had to learn how to live-with *in the platform of your choice*. There is absolutely no excuse for operating systems maligning the user just for the sake of being different from other operating systems. Those that did are gone and forgotten. Then there were those that did an excellent job but couldn't hack the business end of it, such as Commodore who went bankrupt.

> May 14, 1995
>
> Escom AG, who operates 1500 computer stores in Europe, paid $10 million for the rights to the **Commodore** name, its patents and intellectual property in bankruptcy-court action. It plans to resume production of Amiga model in China. It says it will integrate Amiga technology into MSDOS software format used on 80% of the world PCs. Escom has no plans to use any of the former production facilities.

C▷ø *I really meant flip* b
Abort? Retry? Fail? *Abort*
Abort? Retry? Fail? *Fail*
Current drive is no longer valid> ¶
Abort? Retry? Fail? ϡΨ Ωωφη *why not take me back to where I started from?* ξΞεωφΥ Bill Gates; this is 1995.

In the Logical world of computers, oh! why-oh-why do programmers drive users bananas with their illogical quirks. To be cute? or to punish you for the mistake you made? You keep going round-and-round and you get those cryptic sarcastic error reports, *they just didn't do their homework well*. But things have changed a bit to the better and the error codes are more *polite?* and mostly informative.

Nonsense in BASIC

The Timex/Sinclair 2068 reported for VAL error, but

it took the licking and kept on ticking

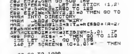
I CAN'T BELIEVE WHAT I SAW!

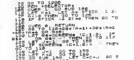
I was trying out some of the software on the ZEBRA Portuguese disk interface and this happened. I LOADed MANAGER, the menu looked quite interesting. However, no matter what key I pressed it was froze in the menu mode. So I pressed break and LISTed the program. Now wait a minute that is not a program - not even a line number. To document it, I LISTed it and to my amazement I got a different output.

I hadn't been drinking although I was tempted to go see if the two bottles of beer were still in the refrigerator, from when the oldest daughter and family were here last summer, but I just sat and finally looked the printout over. Well, this is a Portuguese interface so maybe this is... Ah, but they use letters like we do and this was more like what you'd see in an old Egyptian tomb.

Anybody out there know what went on. I don't even know what the first lines of the program are and have wrote to find out. But still, I wonder if this is an unsolved mystery.

Donald Lambert
Auburn, IN





The Mystery has been reported to Robert Stack of Unsolved Mysteries. But, honestly, it happened to me about 4 years ago when I was fooling around with the Display Modes "dual screen modes" (OUT 255 62 · OUT 255, 2 or something like that). See pages 247/248 of the manual. Anyone out there who can shed some light on what's happening? Editor

What is in LINE 1000?

QXL Notes ♪♫ - Sequel

by Al Tong

Well, I finally got my working copy of SMSQ 2.47 [the QXL's operating system (OS) which mimics the QL's QDOS] — very satisfactory, but not perfect.

First, the excessive keyboard rollover which really made the QXL difficult to use has been corrected. The NUMLOCK can be left 'on' (*good news* if you prefer using the keypad for numeric input)

A spurious character generation still occurs when **TaskMAX** (DRDOS) is used to TASK SWITCH between DOS and SMSQ. Again, this is corrected by pressing the ESCape_KEY and/or ESCape + ConTRoL_E key combination. Since I know how to *correct* the problem on-the-fly, I can live with it. When it becomes too much of a nuisance, I may opt to upgrade to OS/2 WARP.

A minor disappointment is the fact that SBASIC barely accommodates the TURBO compiler. Relatively short and simple programs can be compiled (v3.24); but, long and complex programs return ERRORs which do not occur when compiled on a "regular" QL. I suspect that the LIBERATOR may also have a problem running under SBASIC.

Needless to say, I'll probably be using a 'regular' QL in tandem with the QXL for some time to come until all the irregularities are resolved (if ever).

I keep hoping that another version of the SMSQ OS will be released which better **fills** a VGA screen rather than what I am getting. Now, this may simply be a matter of my not knowing how to configure the single SMSQ module for different video displays. I don't recall seeing any documentation on this (i.e., how to use the 'config' program which is now included); but, that does not mean that it isn't there ... somewhere.

Since my VGA card seems to be able to interpret CGA programs so that they fill the screen as if using a high-res CGA monitor, I naively presume that it would be possible to *re-write* the SMSQ video so that the QL's standard display will also **fill** a standard VGA monitor.

Now, hoping for *Christmas-in-July*, I would like to think that the firmware development that has gone into the soon-to-be-released (probably released by the time this gets into print) MIRACLE MASTERPIECE ENHANCED GRAPHICS CARD will be used on the next SMSQ release. After all, if the MASTERPIECE CARD generates a similarly scaled display on a VGA monitor as the current SMSQ video driver manages, then *that* would surely be a major disappointment. Having more colors in MODE 4 (80-column) is really a trivial enhancement if the display size remains diminutive.

NETworking the QXL

Since my preferred version of TURBO is 2.00, I was originally perplexed when I could not get the NETwork PORTs to function.

It turns out that there is some incompatibility between the SMSQ code and older versions of the TURBO TOOLKIT. I have found that older TURBO TOOLKITs also cause some problems with the MAKE_DIR command, but, MAKE_DIR can be made viable by re-invoking TK2_EXT.

Before the (SUPER)GOLD CARD and the QXL, the only type of NETworking that could be realistically considered was SERVER-CLIENT. This is because a QL is really crippled when FSERVE is invoked to make that particular QL a FILESERVER.

I have found that the QXL and (SUPER)GOLD CARD have sufficient clock speeds that each can be declared as FILESERVERs for the other, thus making a PEER-TO-PEER NETwork a viable reality.

Consequently, my QXL's BOOT program includes the following statements:

```
NET 1
FSERVE
NFS_USE, NDK, N2_flp1, N2_flp2_, N2_ram1_,
N2_ram2_, N2_ram3_, N2_ram4_, N3_ram1_,
N3_mdv1_
```

And, the BOOT for my GOLD CARDed QL includes the following statements:

```
NET 2
FSERVE
NFS_USE WIN, N1_win1, N1_win2_,
N1_win3_, N1_win4_, N1_win5_, N1_ram1_,
N1_ram2_, N3_ram1_
```

You should note two things:

First, if you do not have a hard disk (interface), you can declare up to eight "win()_" devices for the client. The fact that you can have up to eight devices might not be obvious, but it was there in the original, minuscule TOOLKIT2 manual.

Second, if "win()_" is already a device (as on the QXL), then you cannot declare "NFS_USE WIN" since this would conflict with an existing system device name or QDOS keyword.

Since the QXL uses "win()_" devices, my preferred choice is to use "ndk()_" for my NETworked devices (*after* fdk/hdk). "NDK" is an abbreviation for N(etwork)D(is)K.

While you can use any device_label, I quickly found that it is best to use a three-letter abbreviation. If you use a three-letter designation, you can LOAD/SAVE/etc. from within PSION programs (and, most others) along the NETwork. If you choose a non-three-letter designation, the program will probably not recognize it.

DEV_USE has been found to be limited in function. Using DEV_USE to re-label your non-three-letter designation would cripple the NETwork, in part, because functions such as WCOPY will currently return an 'In use' error message. While DEV_USE does allow for LOAD/SAVE, using it to re-label, decreases the overall number of devices you can use.

DEV_USE

You may have read elsewhere that DEV_USE is a **kludge** for older programs that cannot directly access sub_DIRectories. Inevitably, you will want to reconfigure your programs to access a "dev0_" instead of "flp2_" for data files.

My QXL's boot includes these lines:

```
30 DEV_USE 1,WIN2_doc_
31 DEV_USE 2,WIN2_dbf_
32 DEV_USE 3,WIN2_aba_
```

Prosaic — very unoriginal.

As you can see, I have duplicated the suffix as the sub_DIRectory prefix.

Once you reconfigure your programs, you don't have to remember these DEV_USE calls unless you want to access them from the command line prompt.

You can cascade DEV_USE designations, but, I currently don't see the point of it since I have specific sub_DIRectories for specific programs.

FINALLY

Because I often make "_ls" files, I have found it more convenient to have my copy of **Quill** configured so that it still looks to "flp1_" for the **printer_dat** file. Of course, this demands that I remember (!?) to have an *old* start_up disk in the drive (I have forgotten on a couple of occasions, already!).

HAPPY TRAILS,
AND COMPUTING, TO YOU . .

MDIR_BAS v1.05

by Al Teng

MDIR_BAS (v1.05) is a SuperBASIC program designed to facilitate using the MAKE_DIR keyword (FLP/RAM, [SUPER] GOLD CARD, QXL)

The **MAKE_DIR** keyword is used for creating sub_DIRectories — that is, subordinate DIRectories to the main directory. On the QL, these appear as names appended an arrow-type symbol '–>'.

I have found that on some occasions (I think this is caused by using an older TURBO TOOLKIT), the **MAKE_DIR** command is ignored by my QL. This is *corrected* by re-invoking the TK2_EXTensions command.

If you do not yet have the **MAKE_DIR** keyword on your system, then you can modify the program for other purposes by having the appropriate DEFined PROCedures perform the tasks you want.

The SuperBASIC LISTing is not optimized, but, compiles easily.

A CPORTed (ANSI) version is also included for comparison for those interested in seeing how a functional 'C' program looks. I have not tried compiling the MDIR_c code, yet.

Using MDIR_BAS

MDIR_BAS uses the five function_keys and the Escape_key (to quit).

Simply press the appropriate function_key for the QGOS device on which you wish to MAKE_(a_sub_)DIR(ectory). The options are:

flp1_	⇔	[F1]
flp2_	⇔	[F2]
win1_	⇔	[F3]
win2_	⇔	[F4]
other	⇔	[F5]

Thus, if you want to make a sub_DIRectory on 'flp1_' you would press 'F1'. You should see a flashing_cursor within a highlighted (green) strip which should correspond with the function_key that you pressed. INPUT the sub_DIRectory name.

If you INPUT "test" (for example), when you exit the program you should see the name "test –>" in the appropriate DIRectory.

If you INPUT a name longer than 10 characters, the menu window will reset. Use this feature to your advantage if you decide that you have selected the wrong storage device.

If you select 'F5' for 'other', you must input the DEVice name, including the underscore '_'.

Of course, press the ESCape_key if you want to exit the program.

There is no error_trapping in version 1.xx of the program. If you duplicate an existing filename, the program will halt.

(NOTE: MDIR_exe 2.xx has a duplicate name trap and accompanies all PLATYPUS Software programs)

Some thoughts about CPORT & MDIR_c

The 'C' programming language is very much ballyhooed as being the ideal vehicle for writing transportable source code that can then be compiled for different computer operating systems. I don't know if this is true or not.

I have heard that 'C' is apparently a preferred programming language because most "computer science" students have to learn it, and that having gone to the effort, they are reluctant to abandon it. I don't know if this is true or not, either.

I do know that because it is a mid-level language it does not have many of the amenities (i.e., keywords) found in BASIC

I know that there had been many rudimentary concepts about the 'C' language that I did not *grasp* because I had relied quite heavily on various books which obviously presumed you knew things that I obviously did not!

The first was the 'main()' designation. Stating what is now the obvious, it is that portion of a SuperBASIC program that is not contained as PROCedures (presuming you are using PROCedures)

The next thing that is worth mentioning which seemed 'strange' was the structure and notation. Some time ago I ascertained that the 'strange' notation/structure is a byproduct of the language having been designed to be hand written, first, rather than on a terminal. Thus, REMarks are framed by "/* */"; each statement is written on a separate line, and, so on. When scribbled on a legal pad, the notation seems quite appropriate; and, almost logical.

So, I bought my copy of **CPORT** (Digital Precision) from **Mechanical Affinity** last Fall. It was expensive, but it was certainly no more expensive than a course on 'C' would have been, and in the end, there is nothing quite like having the benefit of a *rosette stone* to see how PROCedures and their statements translate directly to the 'C' language from SuperBASIC

Now, my first attempts at using **CPORT** resulted in more ERRORs and WARNINGs than I would have thought.

Initially I was disappointed and frustrated by my first attempts because I use the TURBO compiler which is apparently more demanding than the LIBERATOR, and, certainly more demanding than interpreted SuperBASIC I put the program away for several months since I had other things to do

Well, I come from a long line of read-the-instructions-last users. This is not to say that I don't read the instructions, but since I think the computer is supposed to make things easier, I especially think that most *modern* software should be relatively easy to use. Really good design, regardless of mode of expression, usually has a simple elegance underlying it.

One problem I encountered on my initial attempts to **CPORT** a program was using the INKEY$ keyword. Fortunately there are two, short, sample programs included with **CPORT**, one of which employs INKEY$!

My other problem involved using an array This took more effort, and periodic thought over a six month period.

CPORT's limitations are the SuperBASIC code that you give it to translate GIGO, indeed!

The limitations of the 'C' language's transportability should be obvious when you look at the number of statements which begin 'SB_' (SuperBASIC) suggesting that some *massaging [as Bob Swoger likes to say]* certainly needs to be done to the code if it is to be used on another computer platform

Nonetheless, I hope the inclusion of the CPORTed code makes some aspects of the 'C' language less cryptic

HAPPY TRAILS,
AND COMPUTING, TO YOU ...

MDIR_bas

```
100 DIM BLANK$(24), Knot$(24), th$(20), th2$(15),
    F$(15), t$(3), dev$(24)
110 BLANK$ = "                  " REMark 21 SPACES
120 Knot$ = "mvahd QDOS device!"
130 F$ = "_FLIST_tmp" t$ = "flp"
140 a = 1 POKE 163890, 0 MODE 0
150 Wt: Wz: Wo
160 FILE
170 DEFine PROCedure Wt: WINDOW#2, 512, 256, 0, 0:
    PAPER#2, 7 END DEFine 180 DEFine PROCedure
    Wz: WINDOW#0, 413, 10, 50, 241: PAPER#0, 7
    INK#0, 2
END DEFine
190 DEFine PROCedure Wo WINDOW 462, 250, 25, 3
    PAPER 7 BORDER 1, 7 END DEFine
200 DEFine PROCedure Sound: BEEP 900, 20 PAUSE 5
    BEEP 900, 40 END DEFine
210 DEFine PROCedure Noise BEEP 2000, 20 END
    DEFine
220 DEFine PROCedure CheckKey
230 REPeat key
240 ke = CODE(INKEY$)
250 IF ke = 232 OR ke = 236 OR ke = 240 OR ke =
    244 OR ke = 248 THEN EXIT key 260 IF ke<236 AND
    ke>27 THEN BEEP 900, 40 CheckKey
270 IF ke = 27 THEN Noise PAUSE 10 Sound: Bye.
    EXIT key
280 END REPeat key
290 END DEFine
300 DEFine PROCedure FILE
310 d = 0: CSIZE 0, 0
320 WINDOW#1, 512, 256, 0, 0 PAPER#2, 7 INK#2, 0
    CLS#2 INK 0
330 LINE 0, 96.5 TO 512, 96.5 LINE 0, 92 TO 512, 92
    Noise 340 STRIP 7 INK 2 AT 1, 55 PRINT" @
    PLATYPUS Software' 350 AT 1, 7 STRIP 0 INK 7
    PRINT" DEVICE "
360 MAKER
370 END DEFine
380 DEFine PROCedure MAKER: d = 0
390 WINDOW#0, 124, 132, 42, 20
400 PAPER#0, 7 BORDER#0, 1, 0 CLS#0
410 AT#0, 0, 0 INK#0, 0
420 PRINT#0, \" flp1_ = = {F1} "
430 PRINT#0, \" flp2_ = = {F2} "
440 PRINT#0, \" win1_ = = {F3} "
450 PRINT#0 \" win2_ = = {F4} "
460 PRINT#0 \" other = = {F5} "
```

```
470 STRIP#0, 2 INK#0 7 PRINT#0,\"   MDIR 1.05z   "
    480 STRIP#0 0 INK#0 7 PRINT#0, BLANK$
    INK#0, 0 STRIP#0, 7 490 CheckKey
500 YourChoice = ke
510 SELect ON YourChoice
520 = 232: AT#0, 1, 0 highlight: t$ = "flp" a = 1 dev$ =
    "flp1_" AT#0 1, 0 INK#0, 0 PRINT#0, " ",dev$
    AT#0, 1, 7 INPUT#0, thi$ MakeOne
530 = 236 AT#0, 3, 0 highlight: t$ = "flp" a = 2 dev$ =
    "flp2_" AT#0, 3, 0 INK#0, 0 PRINT#0, " ",dev$
    AT#0, 3, 7 INPUT#0, thi$ MakeOne
540 = 240 AT#0, 5, 0 highlight: t$ = "win" a = 1 dev$ =
    "win1_" AT#0, 5, 0 INK#0, 0 PRINT#0, " ",dev$
    AT#0, 5, 7 INPUT#0, thi$ MakeOne
550 = 244 AT#0, 7, 0 highlight: t$ = "win" a = 2 dev$ =
    "win2_" AT#0, 7, 0 INK#0, 0 PRINT#0, " ",dev$
    AT#0, 7, 7 INPUT#0, thi$ MakeOne
560 = 248 AT#0, 9, 0 highlight: AT#0, 9, 0 INK#0, 7
    PRINT#0, " >" INK#0, 0 AT#0, 9, 2 INPUT#0, thi$
    MakeOTHER REMark use CTRL ]
570 MAKER
580 END SELect
590 END DEFine FILE
600 DEFine PROCedure highlight: STRIP#0, 5 INK#0, 7
    PRINT#0, BLANK$: END DEFine
610 DEFine PROCedure TooLong
620 IF LEN(thi$)>10 THEN Noise MAKER END IF
630 END DEFine
640 DEFine PROCedure MakeOne
650 TooLong
660 IF d = 1 THEN MAKER
670 IF LEN(thi$)<= 10 THEN MAKE_DIR dev$&thi$
680 Sound
690 MAKER
700 END DEFine
710 DEFine PROCedure MakeOTHER
720 IF thi$(5)<>"_" THEN Noise: AT#0, 9, 0 PRINT#0,
    Knot$ PAUSE 30 STRIP#0, 0 INK#0, 7 AT#0, 12, 0
    PRINT#0, Knot$ INK#0, 0 Noise PAUSE 30
    MAKER
730 IF LEN(thi$)>15 THEN Noise MAKER
740 IF thi$(5) = "_" THEN t$ = thi$(1 TO 3) a = thi$(4)
    thi2$ = thi$(6 TO LEN(thi$))
750 IF thi$(5) = "_" THEN MAKE_DIR t$&a&"_"&thi2$
760 Sound
770 MAKER
780 END DEFine
790 DEFine PROCedure Bye CLS#2 INK#2, 2 AT#2, 17,
    28 PRINT#2, "@ PLATYPUS Software" END
    DEFine
```

MDIR_c

```
/*
*       Program  MDIR_c
*       Author   Al Feng
*       Purpose  implement MAKE_DIR keyword
*       CfiXed by CfiX "V4.03" 1995 Apr. 26 18:54:49
*/

#define         prog_version    "1.05z"
#include "Test_h"
#include <eport.h>
#define         BLANK                   "
"
#define         Knot                    "invalid
QDOS device!"
#define         F
        "_FLIST_tmp"
#define         RN                      ""
#define         d                       0
#define         BLANK_ab1               1
#define         dev_ab1                 1
#define         F_ab1                   1
#define         Knot_ab1                1
#define         RN_ab1                  1
#define         t_ab1                   1
#define         thi_ab1                 1
#define         thi2_ab1                1
#define         u_ab1                   1
float a,
float ke,
float YourChoice
CP_FILE sb_channo[16],
char thi[20+1-thi_ab1+1],
char thi2[15+1-thi2_ab1+1],
char t[3+1-t_ab1+1],
char u[16+1-u_ab1+1],
char dev[24+1-dev_ab1+1],

void PROCedure main()  { CP_Initialise(),
        /* "*" DIM stmt deleted - may need to reinitialise
        array(s) */ /*  DIM BLANK$(24), Knot$(24),
        thi$(20), thi2$(15), F$(15), t$(3), dev$(24) */
        /*  21 SPACES  */
        strcpy(t, "flp"),
        a = 1,
        SB_Poke(163890, 0),
        SB_Mode(0),
        Wt(),
        Wz(),
        Wo(),
        FILE(),
        exit(0),
 enddef

void  PROCedure Wt()    /*> 170 <*/
        {
        SB_Window(FNO(2), 512, 256, 0, 0);
        SB_Paper(FNO(2), 7);
 enddef

void  PROCedure Wz()    /*> 180 <*/
        {
        SB_Window(FNO(0), 413, 10, 50, 241),
        SB_Paper(FNO(0), 7),
        SB_Ink(FNO(0), 2),
 enddef
```

```
void  PROCedure Wo()    /*> 190 <*/
        {
        SB_Window(FNO(1), 462, 250,  25,  3),
        SB_Paper(FNO(1), 7),
        SB_Border(1, 7),
 enddef

void  PROCedure Sound()    /*> 200 <*/ {
        SB_Beep(900, 20, 0, 0, 0, 0, 0, 0);
        SB_Inkey(FNO(0), 5),
                SB_Beep(900, 40, 0, 0, 0, 0, 0, 0),
 enddef

void  PROCedure Noise()    /*> 210 <*/ {
              SB_Beep(2000, 20, 0, 0, 0, 0, 0, 0);
 enddef

void  PROCedure CheckKey()    /*> 220 <*/ {
        while ( 1)  {
                /*> key <*/
                ke = (float) (SB_Inkey(FNO(0), 1)),
if (ke = = 232 || ke = = 236 || ke = = 240 || ke = =
244 || ke = =
        248)  {
                        break;
                        /*> key <*/
                endif
                if (ke < 236 && ke > 27)  {
                        SB_Beep(900, 40, 0, 0, 0, 0, 0, 0),
                        CheckKey();
                endif
                if (ke = = 27)  {
                        Noise(),
                        SB_Inkey(FNO(0), 10); Sound();
                        Bye(),
                        break;
                        /*> key <*/
                endif
        endwhile
enddef

void  PROCedure FILE()    /*> 300 <*/
      {
      SB_Cuize(FNO(1), 0, 0);
      SB_Window(FNO(1), 512, 256, 0, 0),
      SB_Paper(FNO(2), 7),
      SB_Ink(FNO(2), 0);
      SB_Cls(FNO(2), 0),
      SB_Ink(FNO(1), 0),
      SB_Line(FNO(1), 0, 96.5, 512, 96 5),
      SB_Line(FNO(1), 0, 92, 512, 92);
      Noise(),
      SB_Strip(FNO(1), 7),
      SB_Ink(FNO(1), 2),
      SB_At(FNO(1), 1, 55),
      fprintf(FNO(1), " @  PLATYPUS Software\n"),
      SB_At(FNO(1), 1, 7),
      SB_Strip(FNO(1), 0),
      SB_Ink(FNO(1), 7);
      fprintf(FNO(1), " DEVICE \n");
      MAKER(),
 enddef

void  PROCedure MAKER()    /*> 380 <*/
      {
      SB_Window(FNO(0), 124, 132, 42, 20),
      SB_Paper(FNO(0), 7),
      SB_Border(FNO(0), 1, 0),
      SB_Cls(FNO(0), 0),
      SB_At(FNO(0), 0, 0),
      SB_Ink(FNO(0), 0);
      fprintf(FNO(0), "\n  flp1_   =  [F1] \n"),
      fprintf(FNO(0), "\n  flp2_   =  [F2] \n"),
      fprintf(FNO(0), "\n  win1_   =  [F3] \n"),
      fprintf(FNO(0), "\n  win2_   =  [F4] \n");
      fprintf(FNO(0), "\n  other   =  [F5] \n"),
      SB_Strip(FNO(0), 2),
      SB_Ink(FNO(0), 7),
      fprintf(FNO(0), "\n   MDIR 1.05z   \n"),
      SB_Strip(FNO(0), 0),
      SB_Ink(FNO(0), 7),
      fprintf(FNO(0), "%s\n", BLANK),
      SB_Ink(FNO(0), 0),
      SB_Strip(FNO(0), 7),
      CheckKey(),
      YourChoice = ke,
switch ( YourChoice ) {

      case 232  SB_At(FNO(0), 1, 0),
          highlight(),
          strcpy(t, "flp"),
          a = 1,
          strcpy(dev, "flp1_"),
          SB_At(FNO(0), 1, 0),
          SB_Ink(FNO(0), 0),
          fprintf(FNO(0), " %s\n" , dev),
          SB_At(FNO(0), 1, 7);
          SB_Input(FNO(0), "%s\n", thi),
          MakeOne();
          break;
      case 236  SB_At(FNO(0), 3, 0),
          highlight(),
          strcpy(t, "flp"),
          a = 2,
          strcpy(dev, "flp2_"),
          SB_At(FNO(0), 3, 0);
          SB_Ink(FNO(0), 0),
          fprintf(FNO(0), " %s\n" , dev),
          SB_At(FNO(0), 3, 7),
          SB_Input(FNO(0), "%s\n", thi),
          MakeOne(),
          break,
      case 240  SB_At(FNO(0), 5,  0),
          highlight(),
          strcpy(t, "win");
          a = 1,
          strcpy(dev, 'win1_"),
          SB_At(FNO(0), 5, 0),
          SB_Ink(FNO(0), 0),
          fprintf(FNO(0), " %s\n" , dev),
          SB_At(FNO(0), 5, 7),
```

```
            SB_Input(FNO(0), "%s\n", thi),
            MakeOne(),
            break,
        case 244  SB_At(FNO(0), 7, 0),
            highlight(),
            strcpy(t, "win"),
            a = 2,
            strcpy(dev, "win2_"),
            SB_At(FNO(0), 7, 0),
            SB_Ink(FNO(0), 0),
            fprintf(FNO(0), "  %s\n" , dev),
            SB_At(FNO(0), 7, 7),
            SB_Input(FNO(0), "%s\n", thi);
            MakeOne(),
            break;
        case 248  SB_At(FNO(0), 9, 0);
            highlight(),
            SB_At(FNO(0), 9, 0),
            SB_Ink(FNO(0), 7),
            fprintf(FNO(0), " \n"),
            SB_Ink(FNO(0), 0),
            SB_At(FNO(0), 9, 2),
            SB_Input(FNO(0), "%s\n", thi),
            MakeOTHER(),
            MAKER(),
            break;
        endswitch
enddef

void  PROCedure highlight()    /*> 600 <*/ {
        SB_Stnp(FNO(0), 5),
        SB_Ink(FNO(0), 7),
        fprintf(FNO(0), "%s\n", BLANK);
enddef

void  PROCedure TooLong()    /*> 610 <*/ {
        if (strlen(thi) > 10)  {
                Noise(),
                MAKER(),
        endif
enddef

void  PROCedure MakeOne()    /*> 640 <*/ {
char cp_strwk1[dflt_str_size], TooLong(),
        if (d = = 1)  {
                MAKER(),
        endif
        if (strlen(thi) < = 10)  {
                SB_Make_Dir(CP_Concat(cp_strwk1, dev,
                thi)),
        endif
        Sound(),
        MAKER(),
enddef

void  PROCedure MakeOTHER()    /*> 710 <*/
        {
char cp_strwk1[dflt_str_size],
char cp_strwk2[dflt_str_size],
char cp_strwk3[dflt_str_size],
char cp_strwk4[dflt_str_size],
        if (thi[5-thi_ab1] ! = '_')  {
                Noise(),
                SB_At(FNO(0), 9, 0),
                fprintf(FNO(0), "%s\n", Knot),
                SB_Inkey(FNO(0), 30),
                SB_Stnp(FNO(0), 0),
                SB_Ink(FNO(0), 7),
                SB_At(FNO(0), 12, 0),
                fprintf(FNO(0), "%s\n", Knot),
                SB_Ink(FNO(0), 0),
                Noise(),
                SB_Inkey(FNO(0), 30);
                MAKER(),
        endif
        if (strlen(thi) > 15)  {
                Noise(),
                MAKER(),
        endif
        if (thi[5-thi_ab1] = = '_')  {
                CP_Shce(t, thi, 1-thi_ab1, 3-thi_ab1), a =
                (float) (thi[4-thi_ab1]),
CP_Shce(thi2, thi, 6-thi_ab1, strlen(thi)-thi_ab1),
        endif
        if (thi[5-thi_ab1] = = '_')  {
                SB_Make_Dir(CP_Concat(cp_strwk4,
                CP_Concat(cp_strwk3,
        CP_Concat(cp_strwk2, t, ftoa(cp_strwk1, a)), "_"),
        thi2)), endif
        Sound(),
        MAKER();
enddef

void  PROCedure Bye()    /*> 790 <*/
        {
        SB_Cls(FNO(2), 0);
        SB_Ink(FNO(2), 2);
        SB_At(FNO(2), 17, 28),
        fprintf(FNO(2), " @ PLATYPUS Software \n"),
enddef

        /* Cport: Translation done, at 2840 statements per
        minute. 0 errors and 0 warnings  */
```

MDIR_h

```
void        Wt(),
void        Wz(),
void        Wo(),
void        Sound(),
void        Noise(),
void        CheckKey(),
void        FILE(),
void        MAKER(),
void        highlight(),
void        TooLong();
void        MakeOne(),
void        MakeOTHER(),
void        Bye(),
```

LIL' AMP RIDES AGAIN!

by Les Cottrell

Dear Bob,

Here is an article for Nite Times News to thank your group for the newsletters and the keyboard article reprints. I am enjoying the big keyboard on my 2068. I have used my son's color ink jet to make key legends, but I haven't finished the mask yet. Function first, then pretty!

After it is published in Nite Times I will probably send it on to Update also. As I said before, you are welcome to publish anything I did for Sinc-Link.

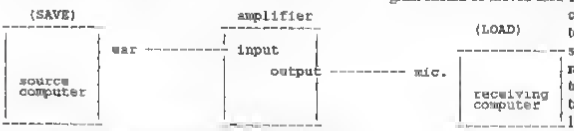
Joan Kealy said you were interested in this write up, so here it is.

If you would rather have inputs on disk rather than hard copy just let me know.

Sincerely Yours,

Les Cottrell January 23, 1995

Many of you started with the ZX-80, ZX-81 or TS1000, TS1500 like I did. (Actually I started with a ZX-80 clone called the MicroAce.) Anyway the point is that we all experienced LOADing difficulties from time to time. One of the cures was to use a little battery powered amplifier from Radio Shack to boost and clean up the signal. I expect many of you have one.



I have recently added a Zebra Timex disk drive to my collection. I needed to transfer some files from the LarKen system to the Zebra and a 3" disk is hardly compatible with a 3 1/2" drive!

I had already learned to move a program from my LarKen computer to a second computer using the LIL' amp. First connect up the source computer ear output to the amplifier input. Then connect the amplifier output to the mic input of the second computer. I have found that about 3/4 volume setting on the amplifier works well, but that may vary for your application. The receiving computer is set up to receive a tape LOAD while the source computer is set up to do a tape SAVE. Start the LOADing sequence on the receiving computer before you start the source computer saving.

If you are moving to a different disk system don't do a SAVE LINE xxx or it will AUTOSTART before you can change the disk commands. Modify any different disk commands prior to doing a disk save of course.

If you are moving a program that has a companion machine code program, check carefully to see which program should be moved first. If you are feeling clever you can set up the command line to do several operations in sequence. I discovered I needed to brush up on the tape LOAD and SAVE syntax.

I have found this to be a reliable way to move from LarKen to Zebra or vice versa. It is better than going from LarKen to tape and then tape to Zebra. It also is a good reminder as to why I went to disk in the first place!

QL Corner

Bob Gilder L.I.S.T.

The 3rd annual North American show will be held on Saturday, June 10th, 1995 in the city of Oak Ridge, TN. QL traders attending the show are Stuart Honeyball of miracle Systems, Tony Firshman of TF Services, Bill Richardson of WN Richardson and Co., Jochen Merz Software, Frank and Carol Davis of Mechanical Affinity and Update magazine, Bill Cable of Wood and Wind Computing plus John Impellizzen and Don Walterman of QBox-USA demonstrating their QL Bulletin board.

It is my understanding that Stuart Honeyball will have the Enhanced Graphics card, the Super Gold Card and QXLs. Tony Firshman will have the Super Hermes, Minerva ROMs and I2C interfaces and Mechanical Affinity will have the QUBIDE Hard Disk Interface plus most of the QL software and hardware from abroad.

Advance registration will be $3.00 or $5.00 at the door. And as usual, a Dutch Treat dinner after the show. Contact IQLR at their North American Office P. O. Box 3991, Newport RI. 02840-0987 or telephone Bob Dyl at 401-849-3805. For additional information call or write to Mel La Verne, 103 Endicott Lane, Oak Ridge, TN 37830-4117, telephone 615-483-4153.

The show will be held at the Faith Luthern Church, 1300 Oak Ridge Turnpike, Oak Ridge, TN. If you plan to stay overnight either before or after the show, the Super 8 Motel, 1590 Oak Ridge Turnpike, Oak Ridge, TN. Reservations by telephone 615-483-1200. Single room rates are $37.00 and double rates are $41.00 and this includes a free Continental breakfast.

I am sorry to say that I will not be able to attend this year's show due to personal circumstances. For all of you in QL land attending this year's show, have a GREAT Time.

The Musical PC8300

by Gil Parrish

The PC8300 is a Hong Kong "clone" of the TS-1000, also known as the "IQ8300" or as "Your Computer". This article is about making music on one, but the info may also be helpful with "your own computer" (no capitals) if your unit is capable of music and you have never tried programming any.

Toward this goal, I do not intend to do a full-blown "review" of the PC8300; extensive articles on it were done years ago, including by our own Don Lambert. But if you are not familiar with it, a basic understanding of this interesting computer may helpful. The PC8300 can use the same peripherals (e.g. RAM expander, printer, and tape drive) as the ZX/TS systems, is able to read ZX/TS tapes, and (having a BASIC which is essentially a superset of ZX/TS BASIC) can run SOME of the same programs. However, the design has a number of changes, some of which seem to have been implimented only for the purpose of avoiding copyright infringment charges, such as differences in memory management (which rule out sharing either machine language programs or BASIC programs which rely on saved variables), a different tape save structure (it can read ZX/TS tapes, but a ZX/TS cannot read its tapes), and perhaps most irritating of all, loss of certain keyboard characters like the quotation mark and the colon key. The keys are not replaced by anything useful (like the apostrophe the ZX/TS lacks), but instead by fanciful graphic characters like a PAC-MAN ghost, a space invader, and a racing car. For this reason, running a program that asks a question ends up printing a sentence with a ghost at the end, which can be disconcerting.

Still, certain changes appear to be genuine improvements. The ivory and green case has a keyboard that is bigger with soft-rubber "chicklet" keys not unlike the later TS-1500. The unit has an extra port for a composite video monitor (and produces a rock-steady light-on-dark picture when used with one), as well as a port for connecting a joystick. The system handles BASIC keywords typed in letter-by-letter; in addition to having a few functions (like PRINT) which can still be punched in with the "one-touch" keyword method if preferred. (I find letter-by-letter a LOT easier.) Finally (and most relevant here), the unit has a built-in speaker with sound capabilities ranging from a "random tone with every keypress" (more than a little distracting at first, but at least you know your keypress was good) through some touted music capabilities supported directly by BASIC.

When I recently acquired one of the units, I started scouting around for programs which would support the unique capabilities of the machine. In this I hit a brick wall. The early attention to the PC8300 seems to have focused on making it more TS-1000 compatible, and indeed, a replacement ROM was even developed for this purpose. Little effort seems to have been put into designing software specifically for the PC8300, although users may have modified a few ZX/TS programs to run on it. So I decided to give custom programming a shot, and, out of the many possibilities (a joystick-controlled program, perhaps? Or a game using those funny graphic characters?), I decided on typing in some music.

I wondered at first if the sound capabilities of the PC8300 might be similar to the later TS-2068, but, a glance at the manuals dispelled that notion. The 2068 supports eight octave three voice music, with control over the waveform envelope, volume and other factors. The PC8300 supports three octave, one voice music, period. So symphonies were out! An additional quirk, the PC8300 manual, like that of the 2068, shows support of "sharps" as well as "natural" notes, but not "flats". This seems to be a factor of having a programmer and not a musician write the manual, since "A sharp" appears to be what most of us know as "B flat". In any event, The 2068, like a regular piano keyboard, has no E♯/F♭ or B♯/C♭. The PC8300 *does* show these keys (E♯ and B♯), whether they could be useful for anything other than irritation of more sensitive ears, I have no idea.

The only advantage of a much less powerful music system is that the commands to run it may be easier too. A command to play an A note on a 2068 might look like--
t0 SOUND 0,124,1,0,8,13,7,62 --followed by yet another command to determine the note duration (like-- 20 PAUSE

60) The command to play an A note in the middle octave of a PC8300 might look like-- 10 MUSIC "A16" --with the "16" representing the duration of the note MUCH easier! For the high octave, a ">" follows the note (e.g., "E>16"), and for the low octave, a "<" precedes the note ("<E16"). Sharps are represented by the same note in reverse video.

Not being a heavy-duty music person, I started my project by heading to the library and getting a book or two of music, mostly simple old tunes and Christmas carols. I chose to start with "Blow the Man Down", that old sea chantey that Hollywood long associated (along with "16 Men on a Dead Man's Chest") with sea pirates. This turned out to be a fairly good choice: not complicated, very repetitive (with seven identical verses and refrains, meaning less programming), and only a few "sharps".

"You'll need to remember: 'Every Good Boy Does Fine'."

If you have no extensive musical knowledge (and I certainly do not), there's a phrase you'll need to remember: "Every Good Boy Does Fine". You see, the musical scale runs from A through G and then back to A, on the treble clef on the top staff (the only ones you will be utilizing for a simple tune, not the chords on the bass clef below), the bottom-most line is E, with the next line G, then B, then D, and at the very top, F. Hence, "Every Good Boy Does Fine" (E-G-B-D-F) helps you identify the notes intersected by the lines. Obviously, a note nestled between the E line and the G line is F, the note nestled below the E line is D: a note just below that-- depicted with a line through it-- is C, and so forth.

But aside from the notes (and the words, since many older, simpler songs may have lots of verses you may never have heard), the important thing the sheet music tells you is the DURATION of the notes. The musical notation runs like:

- 𝅝 = Whole note
- 𝅗𝅥 = Half note
- ♩ = Quarter note
- ♪ = Eighth note
- 𝅘𝅥𝅯 = Sixteenth note
- 𝅘𝅥𝅰 = Thirty-second note

If a dot follows the note, the note duration is increased by 50%. So for instance a dotted half-note would actually be a three-quarter note, if there were such a thing).

To account for the differing note lengths, I assigned a whole note the value of "64", and adjusted all other note values accordingly. Thus a dotted whole note was a "96", a half note "32", a quarter note "16", and so forth down to a thirty-second note "2". I could as easily have assigned the whole note "32" and adjusted all the other values accordingly: that would have worked just as well, unless I happened to run into a dotted thirty-second note (which would have made it "1.5", I have no idea if decimal fractions work or not). But the "64" scale served my needs. Once the notes are programmed, you can insert a "TEMPO" command to pick up the pace. I ended up using TEMPO 5.

Before leaving the sheet music, note that the easiest songs to translate are those that contain no sharp symbols (which looks like "♯") or flat symbols (which look like "♭") to the right of the "𝄞" sign. I will not get into "key signatures", so suffice it to say that a song without such symbols (i.e. in the key of C Major) has no sharps or flats to make things difficult in translation. Music done in different keys will work (and my "Blow the Man Down" falls in that category), but you may not be given clear notice in the sheet music that a particular note is supposed to be a "sharp" or a "flat". My only advice here is that, if a note sounds a bit off, try the "sharp" keys immediately above it and below it. The same note would then be consistently sharp through the rest of the song.

Originally, I intended to have the words displayed on the screen while the tune was running. Unfortunately, I found the screen blanks while playing! Worse, the screen would try to flash back on between lines-- not long enough to be readable, but long enough for the flash to be annoying. I solved this by putting the words on screen for a while before each verse started, and then putting the unit in FAST mode during the song (which seemed to have no effect on the speed of the tune) to prevent the screen flashing back on in the middle. In programming the brief wait between the end of the verse and the beginning of the chorus, I found that I could have the words ("Hey, ho, blow the man down") flashed on the screen by using a PAUSE command or not displayed using a delay loop, my option.

"Unfortunately, I found the screen blanks while playing!"

I fitted all the music and 2 verses of words into the on-board 2K memory. I probably could have fitted in at least one more verse with tighter programming. I saved that program then produced an expanded version using a 16K RAMpack, which had plenty of room for all seven verses!

If anyone with some PC8300-specific programs would like to swap for a copy of my little offering, I can be reached at Route 1, Box 705, Beggs, OK. 74421.

QLuMSi — Recent Chances

by Al Teng

QLuMSi/QL

Using M'cr--S--ft. interface (yes, it's pronounced **clumsy**) is a front-end program, an MS-DOS emulation, and a learning tool. QLuMSi is a TURBO-compiled program that can be EXEC(_W)'d on any QL which has TK2_EXTensions.

For individuals who move between QDOS and DOS, **QLuMSi** allows for a smoother transition when invoking command line input — **QLuMSi** will convert the delimiter for you. Either an *underscore* '_' (the QDOS's standard delimiter) or a *period* '.' (the standard DOS delimiter) may be used within filenames.

For individuals whose spouse may not want to use their QL because they use a DOS system at work, this allows them the comfort of the 'standard' prompts and commands. For example, typing *quill* at an 'A:\>' prompt will instruct your QL to 'EXEC_W flp1_quill' and load the Quill program if it is present on the disk in flp1_.

For individuals who are not familiar with 'standard' DOS commands, it gives them an opportunity to easily familiarize themselves while allowing them to continue to use the QL programs with which they are already familiar.

Where possible, QLuMSi input mimics the MS-DOS command line syntax. Because QLuMSi should also be thought of as a front-end program, some of its functions exceed what can be done from the DOS command line.

Every attempt has been made to ensure **QLuMSi** compatibility with the demands of the MINERVA ROM code as well as the variations found in the SMSQ's SBasic code.

FLIST_imp <==> F(ile)LIST_imp(ort)

QLUSTer and **QLAMBer** users will recognize the FLIST_imp as the dynamic file which contains the names of the sub_DIRectories and files on the medium. The medium must not be write-protected.

The FLIST_imp file can be imported into **Quill** or any other word processor or text editor.

Although QLuMSi 'hides' the FLIST_imp file (on prior versions, the file was named ' '), it is present on the medium.

An ancillary program (deFILE_exe) is included to facilitate removal of the ' ' file by people who are upgrading from earlier versions of the program.

deFILE_exe

The deFILE_exe utility is a truncated version of the QLAMBer front end & utility program. It can be run as a stand-alone program.

SELECT_DEVICE. Use the up/down arrow keys to move the shaded, green bar up-or-down and the left/right arrow keys to change the drive number (1-4). Press ENTER to indicate your choice.

The program will then go directly to the deFILE "page."

Again, use the arrow (cursor) keys to move the highlighting box over the appropriate filename, and then, press the ENTER key.

You will be given an opportunity to abort the process. Press the '(esc)ape' key to exit the program.

COMMANDs

The following COMMANDs are supported (Typing HELP will display this list):

CD\	CLS	COPY	DATE
DEL	DIR	FORMAT	FREE
MEM	PRINT	RENAME	STAT
TIME	TYPE	VER	VIEW

'RD \' & 'MD \' are also supported.

Since 'MD \' accesses the MAKE_DIR keyword (QXL, (Super)GOLD CARD, and FLP/RAM), QLuMSi_COMn (i.e., COMn == common) is amongst the variations provided for systems lacking the MAKE_DIR keyword. The "_COMn" type is the only variation which supports mdv()_:

M == mdv1_ N == mdv2_

Typing "HELP _" (where '_' is the name of a supported command) will reveal more detailed information.

The COMMAND_COM program is the same as the 'full' QLuMSi_COM program except that it lacks the additional HELP beyond the HELP screen which shows the COMMANDs and PATHs which are supported. COMMAND_COM has the advantage of being 'smaller' than the 'full' program.

QLuMSi_EXE is the QLuMSi_COM program which begins with a 'C:\>' prompt (i.e., 'win1_').

Similarly, COMMAND_EXE is the COMMAND_COM program which begins with a 'C:\>' prompt.

VIEW [filename]

COPY [filename] LPT1

Both of these functions now recognize Quill files appended with a "_doc" suffix and generate a roughly formatted screen-or-paper output.

If you VIEW [filename] or COPY [filename] LPT1 the 'UPDATES_doc' you would see something like this:

```
vrm1qdf0
&
H
page nnn

                QLuMSi 4.80 Updates

 Where possible,  QLuMSi Input mimics the MS-DOS
command line syntax.  Because  QLuMSi  should also be
thought of as a front-end program  some of its functions
exceed what can be done from the DOS command line.  To
this end  VIEW and COPY now recognize Quill generated
"_doc" files and generate more usable output.
        VIEW [filename] & COPY [filename] LPT1
 Both of these functions now recognize Quill files appended
with a  "_doc"  suffix and generate a roughly formatted
screen-or-paper output.
 If you VIEW [filename] or COPY [filename] LPT1  this
Updates_doc you would see something like this.
```

and, so on.

Shortly after the end of the document is located, the VIEWing will halt, or a FORM FEED will be sent to the printer.

Although the output is not what you would get using Quill and a printer_dat, it will allow you to look at a screen or paper copy of a document within the QLuMSi program.

Of course, this might seem to be a bit of a kluge for salvaging a corrupted "_doc" file, but, you should be able to salvage all/part of the text of a 'corrupt_doc' file(s) whose formatting data has been corrupted:

COPY [corrupt_doc] lpt1

As before, the non-ASCII characters are 'red' when VIEWed and empty spaces when a COPY is sent to the printer.

If the "_doc" suffix is not present, the program will display the file without the rough formatting.

DIR

DIR /W

An implicit '/p' (for pause) is now presumed. In addition, the standard DIR display now more closely mimics the DOS display:

```
A:\> dir
   Volume in drive A is @ PLATYPUS
   Directory of A:\
Quill              60614 04-02-95 19:33
Archive            52814 04-02-95 19:33
- - - - -
misc ->
turbo ->
printer_dat           85 04-02-95 19:37
GPRINT_PRT           510 04-02-95 19:37
QLAMBer            32330 04-02-95 19:37
QLUSTer            32470 04-02-95 19:37
   Press any key when ready
```

The number of files that are displayed on the screen is dependent on whether the 'banner' is displayed ('CLS' vs 'SHOW').

If you want the DIRectory of another PATH, then type

DIR [path_letter]

To see a DIRectory data displayed in four columns, type

DIR /W

or, DIR [path_letter]:/W

Extra spaces may return a 'Bad command or file name' message.

Some NETwork comments, etc.

First, if your system does not have a hard disk (interface), you may use the eight win0_ devices for NETworking.

Some of you may wonder what and why there are provisions for up to eight (8) ndk0_ devices (snx with QLuMSi_COMn).

While ndk0_ may seem to be an arbitrary NFS_USE designation, it is an abbreviation which stands for n(etwork)d(isk)k [after fdk/hdk].

The reason that there are eight ndk0_ allowances is because you can, in fact, assign that many designations. This is in keeping with what is apparently the standard QDOS device limit of eight.

This quantity may not have been obvious because the original TK2 manual was so tiny that it was difficult to read (of course, that was not the only reason that it was difficult to read!).

In fact, I ascertained the eight device NETwork capability as a what-if supposition, and then went back and re-read the manual to determine whether this had been documented. Of course, it was.

Peer-to-peer NETwork communication, where the QDOS devices have equal access to one another, becomes viable when faster QDOS devices are attached to one another. I currently have a GOLDCARDed QL set up as a 'peer' with a QXL by implementing the following commands in the respective boot programs:

QXL: NET 1
FSERVE
NFS_USE NDK, N2_flp1, N2_flp2_, N2_ram1_, N2_ram2_, N2_ram3_, N2_ram4_, N3_ram1_, N3_mdv1_

GOLD CARD: NET 2
FSERVE
NFS_USE WIN, N1_win1, N1_win2_, N1_win3_, N1_win4_, N1_win5_, N1_ram1_, N1_ram2_, N3_ram1_

The Bottom Line

QLuMSi is available either directly ($15), or as an UPDATE! issue disk ($20; PO BOX 17 MEXICO, IN 46958).

Previous users [direct(CATUG/ZQA!)/EMSoft/UPDATE!] may acquire an upgrade from me ($5/specify disk size).

Al Peng
914 Rio Vista Circle SW
Albuquerque, NM 87105-3324

HAPPY TRAILS,
AND COMPUTING, TO YOU ...

Daisy Be Good - III

by David Lassov

Let's keep looking at Bill Jones' Daisy Disk. Disk No 1 Autoload brings up the version, which is used for input and edit of documents. It is the version, at which we spend most of our time, working at the keyboard. For this, the value of TURBO' is 2.

Two issues ago, we discussed initialization procedures, for putting the 'function' menu on the screen. For those few readers, who missed that, our inputs are 3 2 1 y y y

Last issue discussed item #1 of the 'function' menu, all the way through the 'quicksa' menu. This time we continue with item #2 of the 'function menu, 'Print Header.'

This is a short topic, so we will finish the article, by talking about what happens when our telephone company changes the area code, and what this means to our Daisy software. First, Print Header

With Function Menu on the screen, punch the character 2 on the keyboard. Up comes a request, that you Type the CAP / Ln lgth limit is 80. So we INPUT a caption, whose length is 80 characters, at most. And, the on-line printer puts out a centered header in pica type, before returning to the function menu.

That's all there is to that one, so what does our area code have to do with our Daisy software? (I'm glad you asked!!!)

Well, all the letterheads have to be changed, along with the relevant variables, and do you know how Daisy stores variables?

The letterheads are all derived by Daisy from array n$. n$ is 13 bytes long with n$(9) = 520 884 7667(voice) and n$(12) = 520 882 0388 (data).

Thus, we load Daisy, perform a job BREAK, and ENTER the above values for n$(9) and n$(12). We store this updated version of Daisy, by going to the function menu (GOTO fm); selecting item #5, Data Mgt. Menu (Data Management Menu); going to the save menu, by selecting item #1, SAVE Menu, of the so-called Utility Menu; and, saving Daisy variables and all, by selecting item #1, Save Daisy Pgm + Data, of the SAVE DATA Menu.

That's kinda hard to follow, especially since the system of names, while complete is not entirely consistent. Well, anyone, who has ever heard of Kurt Goedel, can forgive Bill his inconsistencies! But, our job has not been completed either since we have to change the variable file data base too. A catalog of our disk reveals the *vars* file, as 01831 C2. 01831 signifies a variable file of length 1831 bytes, and C2 stands for the extension on all variable files, stored by Daisy. You see, in order to see all the different setups for printer runs simply execute a CAT with a search string of ↑C2. LOADing 01831.C2 into Daisy has the effect of reinitializing Daisy, including all its variables, like n$(9) and n$(12). Now let's update 01831.C2 itself, in order to reflect the new values of n$(9) and n$(12).

First, we LOAD and execute file varset.B6, which sets all variables for Daisy. Next, delete the BASIC file with DELETE, and merge Daisy with RAND USR 100 :MERGE Daisy.B6. Last, go to the SAVE DATA Menu, as above and save the newly updated variable file, by selecting option #E, Save Vars. File Dbase.

Oh yes, don't forget to first update file varset.B6 with appropriate changes to n$(9) and n$(12).

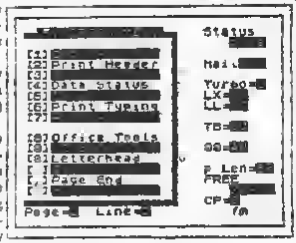
Oh, dear, what about the opening banner? In order to change the telephone numbers there, *as well as* apply color to the screen string, we make use of item #8 Office Tools, of the function menu. We will discuss the office tools menu sometime in the near future.

Well, that's that, but, believe you me, it is easier said than done, due to the special use of the way the 2068 handles its variables and the special advantage we take of that.

Remember how memory-bound the 2068 is as a word processor, only 38k of RAM. Well, Daisy allocates this precious storage to text strings, as much as possible, and uses the rest of its variables, which we're not going to change anyway, into memory starting at *Vars* as contained in 23627. The alternative is to set all variables dynamically, which takes up a lot of additional memory in the program file, beginning at location PROG as contained in 23635.

The Obiger disk operating system has a push-button facility for storing just the variable file, and, when Bill went over to LarKen DOS, he reproduced this facility by clever programming.

Next issue, we will consider item #3 of the function menu, which brings up the Format Menu.



Please consider our new and improved versions, MainAd, PO+MM, IN+ED, and dbase, of Bill Jones' Daisy suite of programs. We LOAD IN+ED, which has 1 for the value of TURBO.

The function menu comes up, directly, as there are no print options and hence no printer initialization.

We choose item #2, Print Header, by punching 2 on the keyboard, and, nothing happens! Instead, we LOAD dbase, which has 0 for the value of TURBO.

The function menu comes right up with no print options or printer initialization. No reaction either to punching 2 on the keyboard.

Ah!, yes, that is a *print option*, and we need either PO+MM (turbo=2) or MainAd (turbo=3).

So, we LOAD PO+MM or MainAd and get to the function menu, by quickly pressing 3, 2, 1, y, y, y, and y, in order to initialize our printer. Punching 2 selects item #2 on the function menu, in order to Print Header. Immediately, we are asked to Type the Caption ":/" LINE length limit is 40. Upon INPUT of a header, no more than 40 characters in length, the on-line printer puts out the caption, *centered* and in *enlarged* Emphasized Pica @ 40 characters per line.

So much for item #2 on the function menu, in order to Print Header.

Please refer to the other article, in order to see how the change in area *code* was effected.

We shall end, by mentioning how delay is avoided, every time Daisy uses LKDOS in its typically a disk-intensive fashion. Daisy spends a lot of its time LOADing those marvelous menus of Bill Jones from disk, and while all menus and supporting software can go onto one floppy disk of 160 tracks, LOADing times are annoyingly slow. But, LOADing all menus from RAMDISK is exciting in its rapidity with a loss of negligible FREE memory. Only 7 blocks of RAMDISK are required.

WAGING IN THE TSROOM

by Donald Lambert

April 8th, 1995 WAG is Wild A__ Guess TSRoom is the room in which I have all my computer equipment setup With the definitions out of the way here we go

When I received the SPRING issue of ZXir QLive ALive! Great except that it always leaves me somewhat depressed since it is an effort by me and there is so little new to me (information that is) Already I am working on the next issue (this material) and Abed has informed me that there is material left over for the next issue

D G Smith has a problem with his LarKen/Oliger disk drive interface. He thought that it might be the fact that he had SAFE v 2.55 but I had used that for a while with both interfaces up and running With Smitty, his problem is that the Oliger will not work if the LarKen is working. So I was testing that out when my computer quit. Well not exactly quit but if either interface was activated at the time the computer was powered up I got ringing bells But if they were both turned off then I could get both to operate I thought that I would swap computers with the one on the AERCO setup but then when I moved it over there the computer would not initialize. But the computer that was on the AERCO would work on the LarKen/Oliger setup. A third computer that was my original computer would work on the AERCO but not if the LarKen board was in the dock port. I was quite puzzled. Then once in a while the computer that I put on the LarKen/Oliger system would give me the bells. I would power down and wait and power up and it would work all right. So I just started to leave the LarKen board switched off just in case.

If I tried to use the LarKen board with the Spectrum EPROM on it on the LarKen/Oliger system the computer would not come up with the initialization. I had known about that for quite a while. Making WAGs I tried to think it through. If it was the computer having been zapped then it would not work part of the time but would fail every time So I wondered if it could be related to the power supply and/or the power circuitry inside the computer RMG has in the resale items a COLECO power supply that is listed as being needed by the AERCO system Is that a hint that the power supply is inadequate? I will have to document my problems and write to Dan Elliott and see if he knows and has a solution for the problems Seems that there is a 5 volt regulator on the computer board that is marginal. If so maybe it needs to be replaced. A belated thought that inforce my thoughts of the power supply during the summer I could not get the computer that failed (the one with the Spectrum EPROM) to initialize Heat? *It sounds like it Don, there was an article in LIST about replacing the little regulator by a larger one*

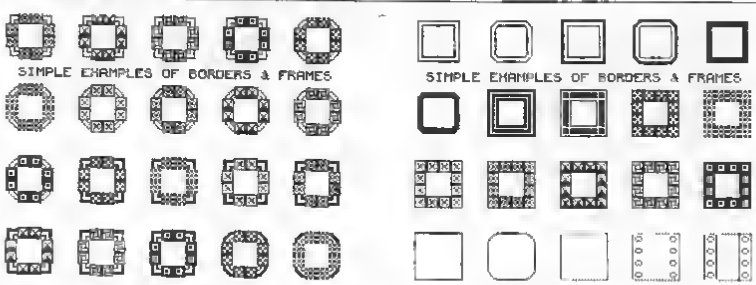
Then there is my learning more about programming I know so little and so I have been typing in the programs from the books that I have (T/S books of course) But nothing is so frustrating than to have a program fail and give some cryptic reason for failure Such as:

2 Variable not found, I510;1

And line I510 is:

1510 PRINT "FILE SIZE "; P-1, "F" , LEN B$

Either 'P' or 'B$' values were not found You have to give them a value Just for a try, in the immediate mode, ENTER, PRINT P and then PRINT B$. One of these will tell you " Variable not found " Once you know which is the culprit, ENTER, LET P= a number or LET B$= a number, to define the variable(s) Run the program or GO TO LINE XXX whatever is the right way Find out in what line the missing variable is and what is supposed to be I don't have enough info to go on, but I assume that this program is something like a word processor "P" might define a PAGE(s) and "LEN B$" might define the length of a line (characters per line) or number of lines per page Editor



by Robert Shade

Past issues are available on disk, via e-mail, or via the Anon-FTP server, garbo.uwasa.fi. The QHJ is always on the look out for

Editor's Forumn

Another few months, another issue. I don't know if there is any special significance to putting out issue #20, but it looks nice to have a zero in the 1's place

One thing nice to report is that the number of QHJ readers is increasing (at least direct readers) I keep running into more QLers on the Internet. I find some and some find me. The e-mail list is now over 120 (mostly outside the US). The hard copy mailing list is in the mid-30's (mostly in the US) I have no idea how many people read the QHJ through friends, BBS's, and archives on the Internet.

One bit of news, there will be a QL show in Oak Ridge, Tennessee, USA, on 10 June. The show has moved from Newport, Rhode Island, where it has been held two times in the past. I plan to attend (it's only about a 5 hour drive for me) I'll bring some QHJ issues, all past issues on disk, my Sinclair Internet Resource list, and disks with Freeware programming languages and tools. Hope to see you there.

I've noticed two things in the latest issue of the International QL Report. Pedro Reina talks about QLIMPO, a C tool that helps write code for different platforms (currently QDOS and MS-DOS, but with an eye out for UNIX) QLIMPO is comprised of 17 objects with 156 functions Pedro has tried to use an Object-Oriented approach to writing this package. QLIMPO is placed in the public domain and all documentation is in Spanish. More details can be found in the March/April 1995 issue of IQLR.

The other is an article by Norman Dunbar entitled "The PE - an idiot's guide!" This is a simple article on the Pointer Environment, what it does, how it works, em Instead of getting into the PE manual, as a beginner this article is a must. This is the best, most detailed, article I have seen on the PE I hope Norman (or someone) would consider putting this article out as a little pamphlet or booklet. It's too good to remain just as an article in a newsletter/magazine. Needless to say, I really liked the article

Program Proposal - Descriptor

In looking through listings of new uploads to the MS-DOS/Windows site CICA, I came up with an idea for a program for the QL. Not having the full expertise or time to work on it, I thought I would present the idea here Hopefully someone will run with it.

Descriptor is a program that allows for each QDOS file to have a "long file name" or description. The best way to describe this program is to show how it will work

When a user is ready to enter a file name into a program (such as Quill), he hits ALT-Q (Query) The Descriptor window pops up The user enters a description of the file (something like a long file name) Examples would be

MFR. Memo to my Boss (MFR stands for Memo For Record) Letter to Mom dated 11/28/94 Article for QHJ on programming

Descriptor would then look up all of the file descriptors that have the string in them. If more than one is found, the user is allowed to choose. The real filename (ie. flp1_text_txt) is then entered into the program (via the keyboard buffer).

The program allows you to almost ignore the real file name for a file and use the long descriptor How it all should work:

Each disk will have a database file (text file) that links file names to descriptors. The file will be called descriptor_db The format will be

filename_ext:descriptor

Descriptor only deems the first 1 m the file as important. This means that colons are allowed in the descriptor

Descriptor will have three functions. Query, Add file, Delete file.

ALT - Q is the hotkey for the Query function.

ALT - A is the hotkey for the Add File function.

Right before saving a file, hit ALT-A, enter the file name and then the descriptor The file name will be entered into the program for you.

ALT - D is the hotkey for the Delete File function.

The user will query for a file (via descriptor) Select the proper file and it will be deleted (out of the database and off the disk) It may be usefull to delete out of the database but not off the disk.

In theory this should work fairly well. In practise, I don't know how well it would do It would help in keeping track of a bunch of text files or Quill files. I've used this type of file naming in a Unix office automation package called Ais (by Applix) It really is much easier to keep track of documents, esp. with lots of memo's and such.

Reverse String

In one of the programming newsgroups I read, I saw a couple of postings dealing with how to reverse a string or a list. A short example would be to take the string "abcde" and make it "edcba" This little puzzle seemed interesting, so I thought I would give it a shot myself.

My first approach is purely iterative. Find the length of the string and then do a FOR loop backwards through the string, adding each character to another string

```
DEFine FuNction reverse$ (in$)
  LOCal rev$, length
  rev$=""
  length = LEN(in$)
  FOR x = length TO 1 STEP -1
    rev$ = in$(x)
  NEXt x
  RETURN rev$
                END DEFine
```

The examples I saw were recursive based, so I thought I would try that approach.

```
DEFine FuNction reverse$ (in$)
```

```
LOCal temp$
if LEN(in$)=1 THEN RETURN in$
temp$ = reverse$( in$( 2 TO ) )
RETURN temp$ & in$(1)
END DEFine
```

How I wrote this reminded me of how I used to do a few Lisp programs. You have to start the procedure with the end condition first. You have to think about how you want the recursion to stop and check for this condition at the start. I then decided to try this program in Lisp using the WS-Lisp interpreter. My first attempt was very similar to the example below. When I was looking at the example code that came with WS-Lisp, I found that it had a reverse function in that example code. I saw that my code was going in the same direction as the example code, but my syntax was lacking. Below is the example code.

```
(de reverse (rev_list)
  (cond
   ((isatom rev_list) rev_list)
 ( t (append (reverse (cdr rev_list))
(list (car rev_list)))) ) ; end of cond
  ) ; end of de revverse
```

Then upon further looking, there was another version of a reverse program that also came with WS-Lisp. It's a bit longer than the first version and not quite as easy to read (at least for me). It seems to rely on the simplest Lisp words. I don't know if it was written to use the lowest level Lisp words or not. Anyway, it's another example to ponder.

```
(de rev (liste)
 (cond
   ( (isnull liste) liste )
   ( (isnull (cdr liste)) liste )
   ( t
    (cons
     (car (rev (cdr liste)))
     (rev
      (cons
       (car liste)
       (rev (cdr (rev (cdr liste))))
      )
     )
    )
   ) ; end of t
 )  ; end of cond
); end of de
```

I'm sure my SuperBasic programs are not the most elegant and can be improved upon. As they said in college, "I leave it as an excerise to the reader."

Gst Qc C Compiler - A Review

I recently found out about another C compiler. Peter Tillier send me a copy (legal, of course) of a C compiler by GST. Peter says that the compiler is now available from Quanta for about 15 pounds.

The compiler seems to be a cross between Small-C and Metacomco QL C. Like Small-C, it supports a subset of the C language, but it supports more than Small-C. Like QL C, it has a compiler, a assembler, and linker. It even uses a link setup file like QL C. And, like C68, even though it has a number of programs to do a full compile, it has a front end to drive the whole process.

In short, QC is based on K&R C and supports switch, for, do, goto statements, logical operators && ||, unary operators ! ~ comma expressions, assignment operators long / short integers, unsigned values, initialised local variables, static and extern single dimension arrays pointers. The preprocessor supports the standard commands, but also supports the inclusion of assembly code. There is a section in the manual that describes how the compiler uses the various 68000 registers.

The standard C function library is supported with more functions that Small-C. QDOS support is complete, although different than some of the other compilers. It does support trap1, trap2 and trap3 (usefull for doing your own tinkering with QDOS).

The manual is fairly complete. It does not give much example code, but it documents the compiler fairly well. The error messages are fully listed and there is even an index.

The compiler fits in between Small-C and QL C (with C68 being far above all C compilers). If you are used to working with Small-C, then QC is a step up in what parts of C are supported. QC provides a greater ability to help in porting than Small-C. QC is not as complicated to use as C68 can be. Sometimes I find the full capability of C68 kind of daunting.

I have not had to really use QC, but from what I can see, I kind of like it. I'm sure I'll always like Small-C, but in those areas that Small-C does not cut it, QC would be a good compiler to use.

Recent Freeware - Apl

Richard Zidlicky has ported a version of APL to the QL. APL stands for A Programming Language. APL is known for being about the worst write-only language. APL uses special symbols as it's operators. This means that it usually requires a special keyboard, thereby making it a language not easy to port. APL is also an interpreted language.

This particular version of APL is based on a freeware Unix version that does not use any special symbols, only the symbols in ASCII. This means that you can not type standard written APL code directly in to this APL. You have to do some converting first.

Since I have only used APL once in College, I really can't say much about this port. It does seem to run with the example code provided with it. I tried to port over some APL code, but I did not know how to translate the funny symbols into the ASCII symbols. As I said, APL is not an easy language to deal with.

Below are a few examples of APL code that came with the interpreter.

```
a{1 2 3C assign a vector to a
b{3 4 5C                      b
a+b C skalar addition
aXb C         mult
a%b C         division
aJ.*b C         inner produkt
now for a few matrix operations
a{4 4R16?17 C random matrix
L<bs>% C invert it
b{I4
```

```
x{bL<bs>&a C solve ax=b
(a+.Xx)-b C .. is that true?
a{5 &R20?21
Ra C now we have 5x4
b{I5
```

APL is designed for matrix operations and is great if you are doing some fairly complex math operations I have a book on computer generated music which is based on APL I'm sure the use of APL is limited in the QL community, but it is always nice to have another language for the QL

This port of APL also comes with some Signal utilities written by Richard. Signals allow communications between processes If this is something you are looking for, then pick up a copy of APL and get the utilities thrown in.

Word Wrap

Now that I have an HP Deskjet 520 inkjet printer, I'm starting to think about what type of output I could do on it. I've found the price of any word processors that support it to be a bit too steep I have rigged up Quill to support one of the fonts built into the DJ520 I would like to use one of the proportional fonts, but Quill (and all text editors) are all monospace based.

I have written a short print filter that supports the DJ520 It supports dot commands (like ROFF or old WordStar) that do things like Bold, Italics, new page, etc

The next step is to add some word wrap facility Below is the source code for a program that does just word wrapping.

It takes in a file, wraps all the words based on the page width (in characters) and outputs the results to another file.

This program is really just a sort test program to focus on how to do word wrapping. By itself it would be rather limited (unless used in a piping environment like Unix) The page width should not be hard coded into the program, but loaded at runtime (either typed in or as a command line argument)

The program expects a few things about the input file It expects a blank line between paragraphs It expects the space, tab, or newline characters to divide words Non-ASCII characters are not handled.

The next step in this program is to add the support of proportional fonts. As is the program treats every character as the same width. With proportional fonts,

characters differ in width (an i is smaller than a w) Once adding proportional fonts is added, then different sized fonts can be added (12 point, 20 point, etc) Output needs to be based on the size of the output (in inches) and not based on the number of characters

```
/* wrap_c
```

This program takes a file as input and reads in each word and reformats the paragraphs based on WIDTH to the ouput file

This program expects a blank line between paragraphs

```
*/
#include <stdio_h>
#define WIDTH   40
main() {
 char  file1[30], file2[30], str[30];
 int   fd1, fd2, temp, length, cur_len;
 printf("Enter Input File Nam      e ;
 gets(file1);
 fd1 = fopen(file1,"r");
 if (fd1 == NULL)  {
printf("Did not open file: %s",file1),
abort(1),
 }
 printf("Enter Output File Name: \n");
 gets(file2);
 fd2 = fopen(file2,"w");
 if (fd2 == NULL) {
printf("Did not open file: %s",file2),
abort(1);
 }
 cur_len = 0;
 while ( 1 == 1) {
       temp = get_word(str,fd1);
   if ( temp == -1)
fputc('\n',fd2); fclose(fd1);
       fclose(fd2);
       abort(0); }
( temp == -2) { fputc('\n',fd2);
fputc('\n',fd2);
      cur_len = 0;
    }
   else {
         length = strlen(str);
  if ( (cur_len + length) > WIDTH) {
       fputc('\n',fd2);
       fputs(str,fd2);
       cur_len = length;
     }
     else {
   cur_len = cur_len + length + 1;
       fputc(' ',fd2);
       fputs(str,fd2);
     }
   }
 } /* end while */
}
/* get_word ( string, file pointer )
   gets the next word in the file. End of a word is space,
tab, or LF. A LF with no word means the end of a
paragraph.
    Return values are:
      0 - no error
     -1 - End of File (EOF)
     -2 - End of Line (EOL)
        (meaning end of paragraph)
*/
get_word( str, fd)
 char str[30],   int fd;
{
 int count, c, lf;
 str[0] = '\0';
 lf = NO;
 count = 0;
 while ( 1 == 1) {
if ( ( c = getc(fd) ) == EOF )  return(-
1); if ( ( c > 32 ) && ( c < 127 ) ) {
str[count] = c;
```

```
        count++;

  /* Space, Tab, CR */
  } ... == 32) || (c == ...) || (c == 10))
        {
        if ( c == 10 )  lf = YES;
if ((count == 0) && (lf == YES))
return(-2); if ( count > 0 ) {
        str[count] = '\0';
        return(0);
        }
```

Recent Freeware - Inform

INFORM is a language used to create text adventure games. It is based on the text adventures that the company INFOCOM used to produce. The adventures were composed of two data files and two programs. The adventure is first written in the INFORM language and compiled with INFORM. This creates a datafile that is then read by ZIP (the adventure interpreter) with runs the adventure. ZIP is available for the QL ported by Luke Roberts. So now with INFORM, the full process can be done on the QL.

The INFORM language resembles C in some respects, but the more purely adventure related words look more like a database programming language (ie fairly verbose). To give you an idea of what the language covers, here are a few chapter titles from the INFORM Language Manual: Objects, Properties and Attributes; Places, Scenery, and the Map; Causing Actions and Making New Ones; Containers, Supporters, and Sub-objects; Doors; Things to Enter, Travel In and Push Around; Living Creatures and Conversation; Starting, Moving, Changing and Killing the Player; Classes of Objects; Adding Verbs and Grammer to the Parser; etc.

The INFORM Language Manual is fairly thick and seems to cover the language fairly well. It's about 100 pages and semi-tutorial, and not just a reference guide. The compiler comes with a number of different sample adventures to learn from and compile.

I have not had a change to give the compiler a spin. I don't know what the demand for text adventures is, but for those interested, it's always handy to have the capability to do what you want. The combination of INFORM and ZIP totally opens up the door to text adventures for the QL. INFORM allows you to compile your own adventures. ZIP allows you to run your adventures or run other adventures from other platforms.

If you are interested in text adventures based on the INFORM language there is a main archive site for such information. It's ftp gmd.de in the directory:

if-archive/infocom/compilers/inform

Object Oriented Programming On The QL

I've been watching the current trend in programming move toward the Object Oriented paradigm for some time now and I still have no idea of what the real differences between Object Oriented Programming (OOP) and procedural program -ming. I have yet to see an article that compares the differences using an example program. To give an example, here is some text describing OOP.

"An object is essentially a black box that contains internal state information. You send an object a message which causes the object to perform some operation. One aspect of an object is that you do not have to know what is inside - or how it works - to be able to use it. From a programming point of view this is very handy. You can develop a series of objects for someone to use. If you need to change what goes on inside, the users of the objects should be unaware."

To me this sounds like someone describing a procedure and not an object. Who really knows the internals of such procedures or fuctions like fopen or getc. You can take out the word object and replace it with procedure and it would still be make sense.

Wanting to try to give OOP a try, I have been looking for a language for the QL that will do some OOP. I found XLISP for the QL. This version of XLISP is XLISP Plus, which has some object oriented features built in.

One of the documents that comes with XLISP Plus is "XLisp 2.0 Object Primer" by Tim Mikkelsen. This document gave an introduction into the object oriented features of XLISP. How Classes, Objects, Messages, and such work. There are a few examples to learn by.

When I first saw this document, I though "Ah, Here is my chance to learn and try out OOP on the QL." Then the reality of learn Lisp hit me. I have been tinkering with Lisp (along with other non-procedural languages like FORTH) for a few years. I must admit that I really can't get the hang of the language. I'm too stuck in my iterative thinking and find it a real bear to read Lisp code. So there goes my grand idea of learning OOP. Besides from what I can gather from the examples, I don't see the advantages it has over procedural programming.

But, for those willing to give it a try the capability is there. If anyone does figure XLisp and OOP out, I hope they will try their best to fill in the rest of us.

While on the subject of Lisp, Scheme (a dialect of Lisp) is also available for the QL. Scheme is an offshoot from Common Lisp (which is what XLisp Plus is based on).

Ohj Freeware Awards

Over the last couple of weeks I have seen a number of award shows, like the Screen Actors Guild Awards, Grammy Awards, Peoples Choice Awards and the Comedy Awards. This started me thinking about awards and lack of them in the QL community. Now magazines, newspapers, and other print media have thier own version of awards. Computer Language magazine has it's Jolt award (named for the soda Jolt - with twice the caffine as Coke). So I think it's time for the QHJ Freeware Awards.

Programmers are an unrewarded lot, especially so for Freeware programmers. Commercial programmers will get monetary compensation. The same goes for Shareware programmers (but even less money and hoping that all users will register the software). But for Freeware programmers, the main emphasis is on free. They do it for the fun on it. Some will write software for themselves and distribute it to others. Some will write software for the challenge of the task. Either way, it's a lot of effort for very little payback.

The QHJ Freeware Awards is designed to recognize the best Freeware programs and programmers over the last year. I've created five different catagories

Best Pointer Environment Freeware Program
Best Non-PE Freeware Program
Best Freeware Port to the QL
Best Freeware Language or Language Utility
Freeware Programmer of the Year

The time for the awards are for 1994. If a program was ported before 1994, but did not make a big impact until 1994, then it can be considered.

I had originally thought about just deciding the winners myself, as some magazines will do. But, I really thought it would be better to get some input from the QL community. My exposure to all the Freeware out there is limited. I could only judge on those that I have tried. Getting input from readers would make the awards truely representative of the QL community.

So, please look over the categories listed above, review what Freeware software you know, and send me your vote for each award. You can send them by mail, e-mail, phone, carrier pigeon, what ever. I will tally the results and report the results in the next issue. Deadline for the votes is 1 May 1995. I hope to have the next issue ready by then. Issue or not, I will make some sort of announcement of some sort at the US QL show on 10 June in Oak Ridge, Tennesse.

I plan to whip up some sort of paper award using Page Designer 3 and my DJ 520 (which means I have to actually learn PD3). I hope to be able to mail the award to each individual programmer that wins.

swensotc@ss2.sews.wpafb.af.mil
tswenson@dgis.dhc.dla.mil

ZQA! Mailing List

Alvin	Alberecht	R Arthur	Gindin	Richard	Norek
Paul	Anderson	Rod	Gowen	J	O'donnell
Gertie	Anderson	Ferdinand	Gunther	John	Olger
Robert	Barnett	William	Harmer	Gilliam	Parrish
Ronald	Baty	Robert	Hartung	John	Pegram
Dave	Bennett	Ted	Heckman	Edwin	Phillips
Don	Berry	Fred	Henn	Hugh	Polley
M H	Binstock	Frederick	Hill	John	Porterfield
Alvin	Bluman	Paul	Holmgren	Harvey	Rait
J A	Bowers	William	Horner	Gene	Ray
Wes	Brzozowski	Leon	Howell	Paul	Robinson
Alex	Burr	Hugh	Howie	Albert	Rodriguez
Bill	Cable	Glenn	Hufstedler	Bill	Russell
Members of	CATUG	Rod	Humphreys	Neal	Schultz
George	Chambers	John	Impellizzeri	Hugh	Scriven
Daniel	Chattin	Warren	Jackson	Robert	Shade
Howard	Chegwidden	Richard	Jelen	John	Shepard
Les	Cottrell	Phillip	Joe	Gregory	Simmons
Larry	Crawford	G David	Johnson	Louis	Simon
Jaime	Cruz-Figueroa	Carl	Jones	Francine	Sklar
Robert	Cumutt	Jon	Kaczor	D G	Smith
James	Curry	Joan	Kealy	Edward	Snow
Frank	Davis	Quentin	Kent	Dane	Stegman
Ed	De Boer	Tezan	Kereluck	Mike	Stephens
Byron	Defries	Wayne	Knaust	Larry	South
William	Des Lauriers	William	Krossner	Alexander	Sweitzer
Jack	Dohany	Jeffrey	Kuhlmann	Tim	Swenson
Dan	Elliott	Donald	Lambert	Jeff	Taylor
Ruth	Fegley	David	Lassov	Jose	Vasquez
Mike	Felerski	Mel	Laverne	Doug	Wagoner
Al	Feng	David	Leech	Stuart	Walton
Mike	Fink	Lafe	McCorkle	Barry	Washington
Charlie	Fox	John	McMichael	Basil	Wentworth
Kenton	Garrett	Harry	Miller	Ivan	Zachev
Robert	Gilbert	Gregory	Newkirk	Wesley	Zapotochna

RMG Enterprises

14784 S QUAIL GROVE CIR
OREGON CITY OR 97045
503 655-7484 10AM - 7PM (Pacific) Tues. - Sat.

Yearly Subscription

Now is the time! Send us 12 #9 or #10 (legal size) self-addressed-self-stamped envelopes and we will send you a pack of information and list of items for sale once a month — for one year.

This List is from some folks who do not need the items! Now it could be yours!

1> TS-2068 Power Supply $20
1> Book-Sams Beginner/Intermediate Manual $5
1> Book-Brain Games $5
Super Special!-- All Of The Above For Only $40 pp

For Above Item(S) Plea.se Use Reference # EWU1291

1> QL Computer w/Peton s/w Coleco p/s, Needs Membrane $55
1> Set Phillips 5.25" 1/2 HT 720K Drives In Case With Power Supply And Cables $125
1> I.C.E. EPROM Cartridge $15
1> Transform Keyboard Cover $5
1> Microbridge s/w -(QL)-FLP $20
1> DBEA.SY on FLP $10
1> Cable C Disk. FLP Filer $15
1> Paste Art V1:FLP $5
1> Qliberator MDV w/Manual $20
20> Update Magazines $25
16> IQLR Magazines $25
18> TIME DESIGNS Magazines $20
1> Taking The Quantum Leap book $20
1> Sinclair QL Computing Book $5
1> QL Survivors Source Book $5
1> QL Icontroller Cursor Controller $25
1> ZX81 16K RAMPAK $5
1> TS Telecommunications Manual 2068 $6
16> Quantum Levels Magazines (QL) $12.50

For Above Items Plea.se Use Reference # CLU1293

1> 6 Pack TS1000 s/w w/Docs ; Includes Shipping! $5
1> Sharp 4602 Laptop, 2 720K Floppy Drives, 640K RAM 10MHZ, 2 Serial/1 Par Ports - RGB/CGA Port - 5.25" Floppy Port - 2 Battery Packs - DOS 3.30 Backlit LCD Screse - Case With Many Pockets Also Included ; All In Like-New Condition Includes AC Adapter
Make Us An Offer! (Rea.sonable!) $ 475 pp

For Above Items Plea.se Use Reference # JCU0989

CNSN 1 All Prices Include Shipping! April 15, 1995

More Items That Would Like To Find A New Home! Rea.d This!

1 QL to PC Monitor adapter (use QL monitor on PC) Like your QL Vision monitor? Want to use it on your PC? This adapter w/ps will allow you to do that with any CGA output. $45
1 TS-2068 In Wooden Case w/Large KB/Reset/More $65 pp.
1 DMP 430 15" Dot Matrix printer (Good Shape) $110 pp

For Above Items Plea.se Use Reference # DSU1190

1 BSR 1200 BAUD External Modem Like New! $25
1 TANDY CoCo Package $35
1 Composite Monochrome 12" Monitor $40
1 TS-2068 w/monitor cable for RGB $65 pp
1 **LarKen** DOS disk package 2 FH drives in case w/PS, LarKen controller and cart. cables $195 pp
1 **LarKen** RAMDISK 256K $125 pp

For Above Items Plea.se Use Reference # REGU0591

1 TS-1000 in KRADLE Keyboard w/added RAM $90
A real Collector's Item!
1 Melbourne House Software Pack 8 Pieces $25
1 SOFTSYNC Software Pack, 9 Pieces $25
1 Software Farms HI REZ Software Pack, 3 Pieces $20
All 3 Titles For The 1000
1 TIMEX Software Pack, 3 Pieces $10
1 Magazine/Book Pack 3 Books/21 Mags $25

For Above Items Plea.se Use Reference # FRSU0792

1 MIRACLE QL Printer Interface $35
1 QL Technical Guide $10
2 AERCO FD 68 Disk Interface w/256K RAM $100 ea.
1 COLECO Power Supply For Use With FD 68 $5
1 RITEMAN 9 Pin Dot Matrix Printer $115
Includes Tractor/Roll Feed w/8 Ribbons Graphics Compatible.

For Above Items Plea.se Use Reference # RDSU0393

1 Timex Sinclair 1000/ZX81 Users Manual $2.50
1 The Timex Personal Computer Made Simple $2.50
1 Mastering Your TS1000 Personal Computer $2.50

Collector's Items

LUPINSKI'S Software Buyer's Guide To TS Products and Services In Loose Leaf Binder $10+3 SH

For Above Items Use Reference # HCU0793

CNSN 3 Last Updated April 6, 1995

We Sure Are Getting A Lot Of These Fine Used Items!

1 PSION Suite S/W Package $17.50
1 AMBER Composite Monitor 12" $45
1 CUMANA QL DISK I/F w/Manual $95

For The Above Items Use Reference # RZU0692

3 TS-1000 8K ROM Chips $3 ea.
3 TS-1000 2K static RAM Chips $3 ea.

For Above Item(S) Plea.se Use Reference # JHU0692

1 TS-2068 Computer w/PS & manual $60
1 TS-2068 Power Supply Only $15
1 ZEBRA OS-64 Cartridge $30
1 OLIGER Serial I/F w/Tape Software $35
2 TS-1000 Computers w/Ps & Cables (Both) $30
7 TS 2968 Cassette Programs (All) $10
20 TS-1000 Cassette Programs (All) $15

For Above Item(S) Plea.se Use Reference # JCU1291

1 OLIVETTI Ink Jet printer PR2300 with 5 cartridges great little printer Ink. carts. are still available from the manufacturer $115